Operations Research Proceedings

Andreas Fink
Armin Fügenschuh
Martin Josef Geiger *Editors*

Operations Research Proceedings 2016

Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Helmut Schmidt University Hamburg, Germany, August 30–September 2, 2016

GOR
Gesellschaft für Operations Research e.V.

Springer

Operations Research Proceedings

GOR (Gesellschaft für Operations Research e.V.)

More information about this series at http://www.springer.com/series/722

Andreas Fink · Armin Fügenschuh
Martin Josef Geiger
Editors

Operations Research Proceedings 2016

Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Helmut Schmidt University Hamburg, Germany, August 30–September 2, 2016

Springer

Editors
Andreas Fink
Institute of Computer Science
Helmut Schmidt University
Hamburg
Germany

Martin Josef Geiger
Institute of Logistics and Organization
Helmut Schmidt University
Hamburg
Germany

Armin Fügenschuh
Applied Mathematics
Helmut Schmidt University
Hamburg
Germany

ISSN 0721-5924 ISSN 2197-9294 (electronic)
Operations Research Proceedings
ISBN 978-3-319-55701-4 ISBN 978-3-319-55702-1 (eBook)
DOI 10.1007/978-3-319-55702-1

Library of Congress Control Number: 2017937263

Printed on acid-free paper

This Springer imprint is published by Springer Nature
The registered company is Springer International Publishing AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This book contains a selection of refereed short papers presented at the *Annual International Conference of the German Operations Research Society (OR2016)*, which took place at the Helmut-Schmidt-Universität/Universität der Bundeswehr Hamburg, Germany, August 30–September 2, 2016. Over 700 participants attended the conference—practitioners and academics from mathematics, computer science, business/economics, and related fields. The scientific program included 475 presentations. The conference theme, Analytical Decision Making, has placed emphasis on the process of researching complex decision problems and devising effective solution methods toward better decisions. This includes mathematical optimization, statistics, and simulation techniques, yet such approaches are complemented by methods from computer science for the processing of data and the design of information systems. Recent advances in information technology enable the treatment of big data volumes and real-time predictive and prescriptive business analytics to drive decisions and actions. Problems are modeled and treated under consideration of uncertainty, behavioral issues, and strategic decision situations.

Altogether 86 submissions have been accepted for this volume (acceptance rate 66%), which includes papers from the GOR doctoral dissertation and master's thesis prize winners. The submissions have been evaluated by the stream chairs for their suitability for publication with the help of selected referees. Final decisions have been made by the editors of this volume.

We would like to thank the many people who made the conference a tremendous success, in particular the members of the organizing and the program committees, the 40 stream chairs, our 12 invited plenary and semi-plenary speakers, our exhibitors and sponsors, the many people organizing behind the scenes, and, last but not least, the participants from 40 countries. We hope that you enjoyed the conference as much as we did.

Hamburg, Germany
December 2016

Andreas Fink
Armin Fügenschuh
Martin Josef Geiger

Contents

Part I
Dissertation Prizes

An Optimal Expansion Strategy for the German Railway Network Until 2030

Andreas Bärmann

Abstract This article summarizes the findings of my Ph.D. thesis finished in 2015, whose topic are algorithmic approaches for the solution of network design problems. I focus on the results of a joint project with Deutsche Bahn AG on developing an optimal expansion strategy for the German railway network until 2030 to meet future demands. I have modelled this task as a multi-period network design problem and have derived an efficient decomposition approach to solve it. In a case study on real-world data on the German railway network, I demonstrate both the efficiency of by method as well as the high quality of the solutions it computes.

1 Motivation

Rail freight traffic in Germany has shown a significant increase over the recent years: it has risen from 291 Mt of transported goods in 2001 to 375 Mt in 2011, a total increase by 29% in 10 years or an average increase of 2.6% per year during this time. These high growth rates are explainable by an overall surge in freight traffic which is due to Germany's continuing economic strength as well as its increasing importance as a freight transit country. From 2011 on, however, there has largely been a sideways trend in the transportation of rail freight. An important reason for this is that many corridors in the German railway network are already operated near or beyond their capacity limits as investment into new capacities has long dragged behind.

This situation as well as the start of the planning process for the new Bundesverkehrswegeplan 2030, the German Federal Transport Infrastructure Plan for the year 2030, have been the motivation for a joint project of the research group Economics, Discrete Optimization and Mathematics at Friedrich-Alexander-Universität Erlangen-Nürnberg and the traffic planning department DB Analytics of Deutsche Bahn AG. This project has been a part of the *KOSMOS* research network funded by the German federal ministry of education and research (BMBF) under the programme "Mathematik für Innovationen in Industrie und Dienstleistungen"

A. Bärmann (✉)
FAU Erlangen-Nürnberg, Cauerstraße 11, 91058 Erlangen, Germany
e-mail: Andreas.Baermann@math.uni-erlangen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_1

(Mathematics for innovations in industry and services) and ran from 2010 to 2013. Its aim was to find out how mathematical optimization can support strategic planning in the expansion of the German rail freight network, which is one of the central questions answered in my doctoral thesis. In its first part, I have developed models and algorithms for an optimal expansion of the rail freight network. The overarching goal was here to be able to transport as much of the forecasted growth in freight traffic until 2030 by rail. This is an important consideration as Deutsche Bahn AG has to reject many transportation orders already now in order to avoid overly long delays on the most-frequented corridors of the network.

2 Modelling the Network Expansion

In my doctoral thesis [1], I have modelled the expansion of the German railway network a as multi-period multi-commodity network design problem, which I state here in a somewhat simplified form. To this end, let us consider the set T, which contains the time steps in the planning horizon, a directed graph $G = (V, A)$, which describes the network, where V are the stations and A the tracks between them, a set of orders $R \subseteq V \times V$, where for each $r \in R$ and $t \in T$ there is a demand of d^{rt} trains to be transported from the origin station $O^r \in V$ to the destination station $D^r \in V$, as well as a set of available infrastructure upgrades B_a for each track $a \in A$. Furthermore, we need the following parameters: the transport costs f_a^{rt} for each origin-destination pair $r \in R$ in time step $t \in T$ on track $a \in A$, the available capacity c_a on each track $a \in A$ and the new capacity C_b that can be created by upgrades $b \in B_a$ on track $a \in A$. If we denote by $B = \cup_{a \in A} B_a$ the set of all available infrastructure upgrades, the problem can be described by a mixed-integer program (MIP) with the following structure:

$$
\begin{aligned}
\min \quad & \sum_{t \in T} \sum_{r \in R} \sum_{a \in A} f_a^{rt} y_a^{rt} \\
\text{s.t.} \quad & \sum_{a \in \delta_v^+} y_a^{rt} - \sum_{a \in \delta_v^-} y_a^{rt} = \begin{cases} 1, & v = O^r \\ -1, & v = D^r \\ 0, & \text{otherwise} \end{cases} \qquad (\forall t \in T)(\forall r \in R)(\forall v \in V) \\
& \sum_{r \in R} d^{rt} y_a^{rt} \leq c_a + \sum_{b \in B_a} C_b u_b^t \quad (\forall t \in T)(\forall a \in A) \\
& u \in U \\
& u \in \{0,1\}^{|T| \cdot |B|} \\
& y \in [0,1]^{|T| \cdot |R| \cdot |A|}.
\end{aligned}
$$

In this model, variable u stands for the upgrades which are realized up to a certain year, and y represents the routing of the trains in the network. The objective function then minimizes the transportation costs (and thus maximizes the profits), while the routing of the trains has to fulfil flow conservation, and the available capacity must be respected on all the tracks. The constraint $u \in U$ models further requirements for

a feasible expansion, such as keeping to a certain budget in each year of planning and compatibilities among the available infrastructure measures. This problem is a typical network design problem, except for the fact that it additionally asks for a schedule for the expansion of the network: which upgrades are to be realized in which time intervals?

3 Solution of the Problem via Decomposition

In the literature, there are severable well-known methods to solve network design problems of different kinds in an efficient manner, such as Langrangean relaxation, Benders decomposition or Dantzig-Wolfe decomposition, where the latter leads to a path-based formulation. For the problem considered here, all these methods have certain drawbacks, which is due to the size of the underlying graph—1600 nodes (stations), 5200 directed arcs (tracks) and 3600 origin-destination pairs, which has to be taken times 20 to reflect the planning horizon. The subproblems in the above solution approaches (single- or multi-commodity flow problems) thus have to be solved on very large graphs and typically have to be solved very often. This was the motivation to develop a new decomposition method that delivers high-quality solutions already after few subproblem evaluations. It is based on the idea of [2] to solve multi-period network design problems by first determining a suitable target network via an ordinary single-period network design problem and then finding a favourable year-by-year plan to install the new capacities. The latter is done in [2] by using single-period network design problems to do an iterative backwards elimination of capacities to get from the target network to the initial network, in each period removing capacities worth at most the given budget for each time step. The disadvantage of this backwards elimination in terms of solution quality is that it takes a very local view onto the planning horizon—it always passes from one year to the next. Therefore, I have developed a method named *multiple-knapsack decomposition* in my dissertation which replaces this second step by a scheduling subproblem based on estimations of the benefit of having each upgrade in place in a certain year of planning. It is of the form:

$$\begin{aligned} \max \quad & \sum_{t \in T} \sum_{b \in B} \mu_b^t u_b^t \\ \text{s.t.} \quad & u \in U \\ & u \in \{0,1\}^{|T| \cdot |B|}, \end{aligned}$$

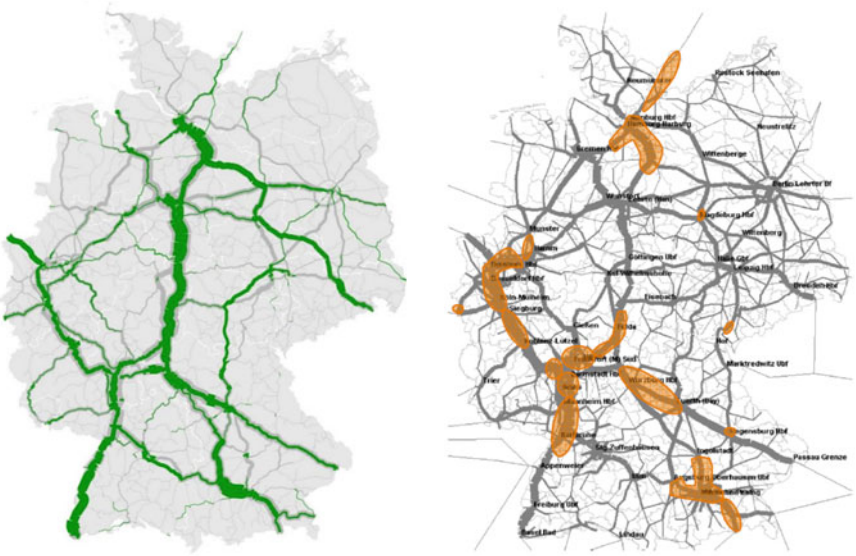
where the parameter μ is an estimation of the cost reduction that is made possible by the infrastructure upgrades (shorter transport paths or less rejected orders). It is calculated from the utilization of the new capacities that have been determined as candidate upgrades in the first step, for which we solve a series of single-period multi-commodity flow problems. These utilization values are then used to estimate the contribution of each upgrade to the total reduction of transportation costs.

The above scheduling problem takes the form of a multiple-knapsack problem with further constraints if there is a budget restriction for the upgrades as is the case here, thus the naming of the method.

This approach has led to very good results in the computational study conducted together with DB Analytics: the deviation from the optimal solution did not exceed 2% for any of the considered instances (subnetworks of the German railway network), exhibiting much shorter solution times than an MIP solver. For the complete Germany-wide instance that took 24 h to solve without decomposition on a compute server with 500 GB of memory, multiple-knapsack decomposition only took 8 hours on a workstation with 64 GB of memory and much weaker processors. By suitable parallelization, the solution time could be reduced to about 20 minutes, as the subproblems estimating the benefits of the upgrades in each year of planning can be solved independently of each other. The derivation from the optimal solution was only 0.78%, which makes the developed method very suitable as a quick heuristic for a planner who would like to evaluate several different demand scenarios within short time. As I could show in addition, the presented decomposition heuristic can be extended to an exact solution approach by a suitable embedding into a Benders scheme if needed.

4 A Case Study for the German Railway Network

Together with my doctoral supervisor, Alexander Martin, and our contact person at Deutsche Bahn AG (DB), Hanno Schülldorf, I have conducted an extensive case study for the necessary infrastructure upgrades in the German railway network. Besides the data on the railway network, it was based on the company-internal demand forecasts in rail freight traffic until 2030, which are depicted in Fig. 1. Figure 1a shows the expected cumulative growth in freight and long-distance passenger traffic on the tracks of the German railway network between 2010 and 2030—a thicker green marking indicates a higher growth on a certain track. In Fig. 1b we then see on which tracks DB expects bottlenecks by 2030 as a result of this growth if there is no expansion of capacities in the German railway network. Partly, the marked line segments are at their capacity limit already today, as indicated in the introduction. Altogether, the DB-forecast for the growth in rail freight traffic expects an increase in the number of operated trains per day of 50% until 2030 as compared to 2010, i.e. about 2% per year on average. In our study, we have considered 4 different types of infrastructure measures to increase the capacities of existing links in the network (construction of new lines was not part of the study): laying new tracks, speed-improving measures (such as the creation of overtaking facilities), the electrification of diesel lines and block size reduction (reduction of the necessary safety distance on a track by an improved train control). In the order as mentioned above, these measures are decreasing with respect to the new capacity they create on a given link, the financial investment necessary as well as the time of implementation. This requires a weighing of cost and benefit which also takes into account



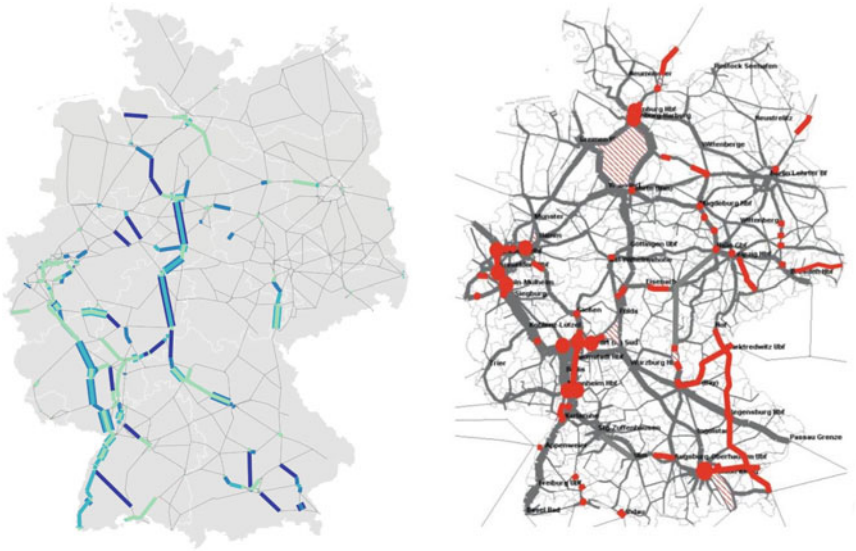
(a) Predicted cumulative growth of freight and long-distance passenger traffic on the main corridors between 2010 and 2030

(b) Expected bottlenecks in the railway network in 2030 without the creation of new line capacities

Source DB Netz AG (2013)

Fig. 1 Visualisation of the DB-forecast for the growth in railway traffic until the year 2030. *Source* [4]

how fast the desired effect can be achieved. The models and algorithms we have developed have allowed us to provide Deutsche Bahn AG with a planning software that now enables network planners to evaluate different options for an expansion of the railway network, depending on the assumed demand scenario and the planned budget. For the scenario shown in Fig. 1a and an annual budget of 700 million Euros per year, for example, we obtain the expansion plan depicted in Fig. 2a in four construction phases. Based on this solution, it is not only possible to analyse the chosen upgrades, but also to evaluate the underlying traffic flows in the railway network, which allows to check the plan for plausibility. For example, the plan shown in Fig. 2a does not yet take into account the classification of lines through central Germany as highly mountainous, which therefore exhibit much higher costs for the same type of upgrade. When Deutsche Bahn AG now prepares the input data to reflect this, it is to be expected that the proposed solution tendentially leads around the mountainous passages—as the official expansion strategy in the Bundesverkehrswegeplan 2030 does it. It is shown in comparison in Fig. 2b. According to the opinion of planners at Deutsche Bahn AG, the described approach is in any event a very good starting point for an optimization-supported network planning. More details on our decomposition approach and the case study for the German railway network can be found in [3].



(a) Our expansion plan in four upgrade phases until 2015, 2020, 2025 and 2030 respectively (darker colour = later year)

(b) The offical expansion strategy by DB Netz AG – upgrades are marked in red. *Source* DB Netz AG (2013)

Fig. 2 Comparison of the upgrades chosen in our study and the official plan put forward by Deutsche Bahn AG for the Bundesverkehrswegeplan 2030

References

1. Bärmann, A.: Solving network design problems via decomposition, aggregation and approximation—with an application to the optimal expansion of railway infrastructure. Ph.D. thesis, Friedrich-Alexander-Universität Erlangen-Nürnberg (2016)
2. Minoux, M.: Network synthesis and optimum network design problems: models, solution methods and applications. Networks **19**(3), 313–360 (1989)
3. Bärmann, A., Martin, A., Schülldorf, H.: A decomposition method for multi-period railway network expansion—with a case study for Germany. Transp. Sci. (2015) (to appear)
4. DB Netz AG: Netzkonzeption 2030: Zielnetz der DB Netz AG für die Schieneninfrastruktur im Jahr 2030. Information booklet (2013)

On- and Offline Scheduling of Bidirectional Traffic

Elisabeth Lübbecke

Abstract This work summarizes insights related to bidirectional traffic on a stretch containing bottleneck segments. On a bottleneck segment, concurrent traveling of vehicles in opposite direction is restricted. The considerations are motivated by the ship traffic at the Kiel Canal which connects the North and Baltic Seas and is operated bidirectionally. Since ships register their travel requests only on short notice, we investigate the Canal's ship traffic additionally in the online setting.

1 Introduction

We consider bidirectional traffic where bottleneck segments restrict concurrent traveling of vehicles in opposite direction. Single tracks in railway planning are an example for such bottlenecks. This work summarizes results of [8] that considers the example of ship traffic at the Kiel Canal.

Situated in the north of Germany, the Kiel Canal connects the North and Baltic Seas. With more passages than the Panama and Suez Canals together, it is the world's busiest artificial waterway. Compared to the way around Denmark, the canal saves an average of 250 nautical miles (460 km). The Kiel Canal, as the more ecological and safer route, became the basis for the trade between the countries of the baltic area with the rest of the world [7].

Since offshore vessels are not primarily designed for inland navigation, the passing of two ships with large dimensions is not possible at arbitrary positions. To facilitate bidirectional operation of the Kiel Canal, wider areas within the canal called sidings are needed that allow for passing and waiting, see Fig. 1. This yields a sequence of bottleneck segments and decisions must be made about who is waiting for whom, where, and for how long. Responsible for these decisions is the Waterways and Shipping Board (WSV/WSA) with a team of nautically experienced expert navigators.

E. Lübbecke (✉)
Workforce Management Division, INFORM GmbH,
Pascalstr. 23, 52076 Aachen, Germany
e-mail: elisabeth.luebbecke@inform-software.com

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_2

They try to keep the necessary waiting times in sidings on average over all ships as small as possible.

In expectation of a tremendous continuing growth of traffic demand an enlargement of the canal was planned. There are a bunch of possible construction options such as extending or creating sidings or to allow more flexible passing of ships by deepening and/or widening crucial parts of the canal. In order to assess the cost and benefit of these options their combined effects under predicted ship traffic needed to be reliably estimated. To that end, we developed an optimization tool for the "Planungsgruppe für den Ausbau des Nordostseekanals" of the WSV that emulates the current ship traffic control. This tool was used to evaluate the various construction options with the aim of selecting a most adequate combination.

In addition to the developed ship traffic control tool, insights being relevant beyond the concrete scope of the Kiel Canal are provided. These investigations concentrate on two characteristic properties of this ship traffic control. First, its *bidirectional* component is investigated in further detail. Second, we account for the fact that decisions must be adapted *online* since ships register their requests only shortly before their arrival.

2 Bidirectional Scheduling

For the theoretical investigations, we discuss the problem's natural relation to classical machine scheduling and analyze similarities and differences. This analysis concentrates on the following characteristic property being common for all kinds of bidirectional traffic on bottleneck segments: vehicles moving in the same direction can enter a tight lane sequentially with relatively little headway while vehicles in opposite direction must wait until the whole lane is empty again, cf. Fig. 1. With a compact scheduling model that accurately accounts for this specialty, a detailed analysis of the problem's off- and online complexity is accomplished. It facilitates the development of algorithms with provable performance bounds on the one hand and the identification of hardness inducing properties on the other hand.

Having flow shop scheduling in mind, we generalize rectangle jobs, that are passing through a sequence of machines in the given order, to parallelograms to be arranged on a sequence of segments in the given or opposite order, cf. Fig. 2. By that, a job is represented by two values on the time-axis: the time spent for entering

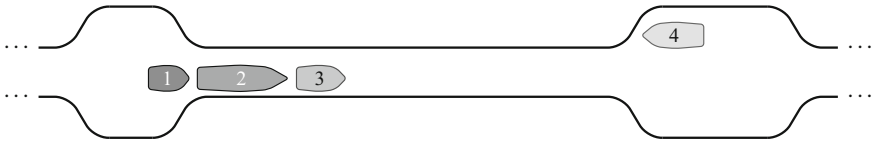


Fig. 1 Bidirectional ship traffic at the Kiel Canal. Ship 4 must wait in a siding until ships 1, 2 and 3 have left the bottleneck segment

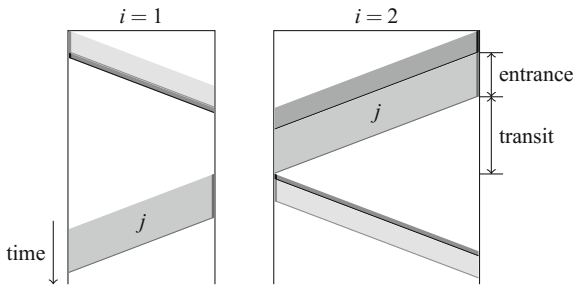


Fig. 2 Example of the compact bidirectional scheduling model with parallelograms representing movement of vehicles from *left* to *right* and *right* to *left*

the segment (the time between the first entrance of the prow and the moment the stern has accessed the segment) and the additional time needed to traverse the segment after the entering is finished. While the former prevents the segment from being used by any other job (running in *either* direction), the latter only blocks the segment from being used by jobs running in *opposite* direction. An intersection-free arrangement of parallelograms with appropriate orientation then ensures a feasible (collision-free) movement of vehicles where those with equal travel-direction can use a segment concurrently. Therefore, packing parallelograms relaxes the restriction that a resource can only be used by one job at a time since it is extended by a second dimension corresponding to positions.

With rectangles being special parallelograms, hardness results from scheduling carry over to the bidirectional case. From the application point of view the time for transit is dominating the time for entrance. Hence, we are especially interested in difficulties that additionally arise from delays induced by orientation-switches of the parallelograms instead of varying entrance times. To that end, we fix these entrance times in our considerations. By this we get insights according to complexity in dependence of the number of considered segments. Furthermore, the bidirectional traffic at the Kiel Canal is distinguished by the specialty that exceptions for the passing of ships with smaller dimensions exist. Thus, we consider parallelograms with opposite orientation that are *compatible* and hence, are allowed to overlap. In our investigations, we complement NP-hardness for general compatibilities with a classification of compatibilities that admit efficient exact algorithms. In the case of entrance times that are not fixed, the techniques of [1] can be extended to prove the existence of a polynomial time approximation scheme (PTAS) for bidirectional scheduling on a single segment. For further details on the complexity results we refer to [2].

In the online setting, we are interested in the increase of the costs due to the circumstance that vehicles register their transit requests only on short notice. In an online instance of bidirectional scheduling, jobs are not known in advance but appear by their release date. Once, an online algorithm has started a job on a segment it is not possible to revert the decision since it corresponds to the movement of the corresponding vehicle. In our considerations, we apply the common technique of competitive analysis where the results of an online algorithm over all instances are compared to the optimum an offline algorithm can achieve. This comparison is quantified by the *competitive ratio* and finding the best possible one yields a meaningful

measurement of the loss caused by the online restriction. For bidirectional scheduling in general, we can bound the best possible competitive ratio from below by 2 and from above by 4, compare [5, 6]. For special cases, we can provide polynomial running time and decrease the gap between lower and upper bounds on the best possible competitive ratio. However, as in many online optimization problems we are not able to close this gap.

3 Competitive-Ratio Approximation Schemes

The concept of competitive-ratio approximation schemes is an alternative approach to deal with such open gaps in online scheduling [4]. Such schemes compute online algorithms with a competitive ratio arbitrarily close to the best possible competitive ratio. To that end, a new way of designing online algorithms is presented for the example of parallel machine scheduling with preemptive and non-preemptive jobs to minimize the weighted sum of completion times. The approach can furthermore be extended to bidirectional scheduling for a single segment.

In addition to structuring and simplifying input instances as in [1], an abstract description of online scheduling algorithms is used to reduce the infinite-size set of all online algorithms to a relevant set of finite cardinality. In addition, the competitive ratio of these algorithms can be approximated with $1 + \varepsilon$ precision. This combination is the key for eventually allowing an enumeration scheme that finds an online algorithm with a competitive ratio arbitrarily close to the optimal one and that approximates the corresponding value up to a $1 + \varepsilon$ factor. This implies a respective estimate for the optimal competitive ratio.

The approach differs strongly from those where (matching) upper and lower bounds on the competitive ratio of a particular and of all online algorithms were derived manually. Instead, the search for the best possible competitive ratio for the considered problems can be tackled by executing a finite algorithm.

4 The Ship Traffic Control Problem

For the original problem of ship traffic control at the Kiel Canal [9] more complex feasibility constraints for instance in the sidings must be respected. To that end, we combine the scheduling perspective with a dynamic routing approach and therefore integrate algorithmic ideas from two important related applications, train scheduling on a single-track network [11] and collision-free routing of automated guided vehicles [3]. The idea is roughly, to embed a sequential (local) routing method which considers only one ship at a time, in a simultaneous (global) scheduling method to optimize the complete fleet. In addition, we embed the algorithm into a rolling horizon approach. It is implemented such that after new request-information is incorporated, all parts of the solution that are not fix by that point in time is reconsidered again.

The sketched algorithmic combination yields a fast heuristic with an average running time of less than 2 min for historic instances covering a time horizon of 24 h each. The achieved objective function values significantly improve upon manual plans. Results of smaller instances are compared with instance-dependent lower bounds. The calculated solutions have been presented to the expert planners by animated ship-movements and as interactive distance-time diagram, see Fig. 3. The latter is based on the diagram that is used by the planners on site who approved the presented solutions. Most importantly, the practical context is modeled in such a high level of detail that the resulting tool perfectly reflects the effects of enlargement options at the Kiel Canal. This enabled the officials to evaluate the different options under ship traffic predicted for the year 2025, and to base their decisions on the simulation results.

Even though it was not intended by the study, the heuristic complies with important requirements for computer aided traffic control. In fact, the planning in rolling horizons is able to deal with the present online character. It perfectly integrates with a further heuristic [10] that schedules the locking process at each boundary of the Kiel Canal since entering, passing, and exiting the canal are interdependent. The overall system may support the expert planners during several potentially difficult years of construction work. Moreover, it was considered to use the tool for deciding

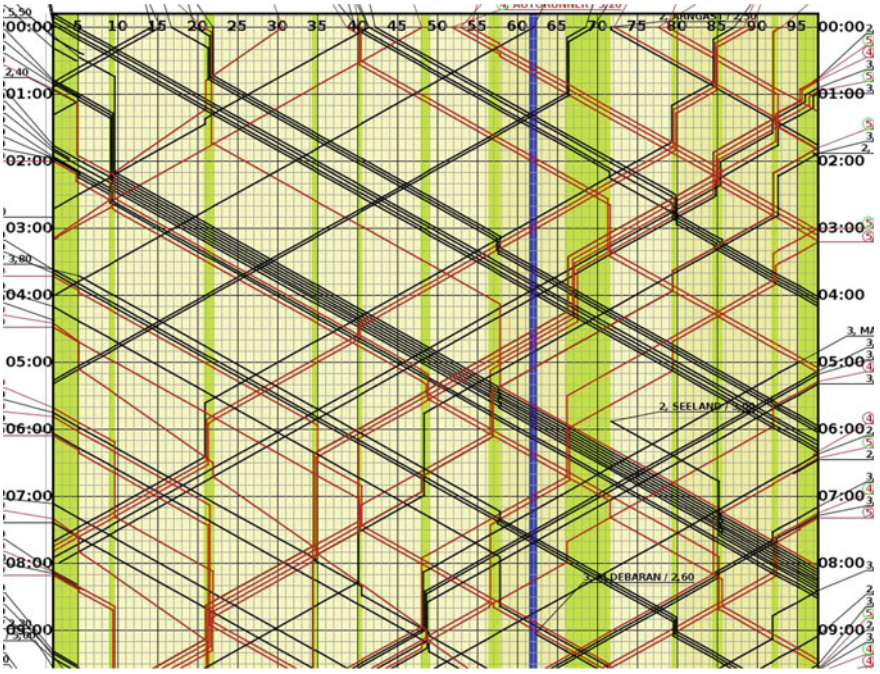


Fig. 3 A distance-time diagram presenting a solution of a complex ship traffic control instance that was calculated by the developed heuristic

about the schedule of the construction work itself: Different orders of construction and the selection of different excavating machines (several of which significantly hinder regular traffic) directly influence the traffic flow under scarce resources.

5 Summary

To summarize, we implemented a solution of high practical value that is able to tackle bidirectional traffic (1) as offline heuristic solving many instances of reasonable size fast and sufficiently detailed for meaningful study-results and (2) as online tool which is applicable as suggestion tool for the daily planning. In addition, we provide a compact model that admits theoretical insights on the nature of bidirectional traffic. It emphasizes the challenges occurring at bottleneck segments where bidirectional operation is necessary since they are origins of large delays. The investigations are accomplished in the offline and the online case. Finally, we present a new approximation concept for competitive analysis in online scheduling.

Acknowledgements The thesis [8] was supervised by Rolf H. Möhring and Nicole Megow. It was written at the Department of Mathematics, Technische Universität Berlin and supported by the DFG Research center MATHEON *Mathematics for key technologies* in Berlin.

References

1. Afrati, F.N., Bampis, E., Chekuri, C., Karger, D.R., Kenyon, C., Khanna, S., Milis, I., Queyranne, M., Skutella, M., Stein, C., Sviridenko, M.: Approximation schemes for minimizing average weighted completion time with release dates. In: Proceedings of 40th Annual Symposium on the Foundations of Computer Science (FOCS), pp. 32–43 (1999)
2. Disser, Y., Klimm, M., Lübbecke, E.: Scheduling bidirectional traffic on a path. In: Proceedings of 42nd Colloquium on Automata, Languages, and Programming (ICALP), vol. 9134, LNCS, pp. 406–418 (2015)
3. Gawrilow, E., Köhler, E., Möhring, R.H., Stenzel, B.: Dynamic routing of automated guided vehicles in real-time. In: Krebs, H.-J., Jäger, W. (eds.) Mathematics: Key Technology for the Future, pp. 165–177. Springer, Berlin (2008)
4. Günther, E., Maurer, O., Megow, N., Wiese, A.: A new approach to online scheduling: Approximating the optimal competitive ratio. In: Proceedings of 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 118–128. SIAM (2013)
5. Hall, L.A., Shmoys, D.B., Wein, J.: Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. In: Proceedings of 7th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), vol. 96, pp. 142–151. SIAM (1996)
6. Hoogeveen, J.A., Vestjens, A.P.A.: Optimal on-line algorithms for single-machine scheduling. In: Proceedings of 5th International Conference on Integer Programming and Combinatorial Optimization (IPCO), vol. 1084, LNCS, pp. 404–414 (1996)
7. Kiel Canal Official Website. http://www.kiel-canal.org/english.htm (2016). Accessed Nov 2016
8. Lübbecke, E.: On- and Offline Scheduling of Bidirectional Traffic. Ph.D. Thesis, TU Berlin. Logos Verlag Berlin GmbH (2015). ISBN 978-3-8325-4115-6

9. Lübbecke, E., Lübbecke, M.E., Möhring, R.H.: Ship traffic optimization for the Kiel Canal. Technical Report 4681, Optimization Online. http://www.optimization-online.org/DB_HTML/2014/12/4681.html (2014)
10. Luy, M.: Algorithmen zum Scheduling von Schleusungsvorgängen am Beispiel des Nord-Ostsee-Kanals. Master's Thesis, TU Berlin (2010)
11. Szpigel, B.: Optimal train scheduling on a single track railway. In: Ross, M. (ed.) Operational Research '72, pp. 343–352. North-Holland, Amsterdam (1973)

Integrated Segmentation of Supply and Demand with Service Differentiation

Benedikt Schulte

Abstract The presented research addresses the integrated segmentation of supply and demand with service differentiation by means of service-level menus. To this end, it establishes a joint perspective on the market side—that is, prices and service levels—and the operations side—that is, the inventory management policy and the corresponding parameters. This joint perspective comprises analyzing when the introduction of a service-level menu increases profits over those of a single undifferentiated offering and how to design optimal service-level menus. Surprisingly, in many cases service differentiation does not increase profits significantly. One way to interpret this finding is that differentiating customers based on service levels alone is a weak differentiation lever only, that is, the price differences between offerings with differing service levels need to be small in order to prevent customers from switching to offerings with lower prices and service levels. Therefore, successful price differentiation requires service differentiations being supported by presence of additional conditions or measures (e.g., pricing restrictions or further differentiation levers). Indeed, it is possible to show that service differentiation can significantly increase profits if the company experiences pricing restrictions.

1 Introduction

This article presents the author's Ph.D. thesis [3], which has been awarded the dissertation price of the German Operations Research Society (GOR) during the OR conference 2016 in Hamburg. This thesis addresses the integrated segmentation of supply and demand with service differentiation by means of service-level menus. The following sections provide an introduction to the topic of service differentiation (Sect. 2) and an overview of some of the thesis' main findings (Sect. 3).

B. Schulte (✉)
Chair of Logistics and Quantitative Methods in Business Administration,
Würzburg University, Sanderring 2, 97070 Würzburg, Germany
e-mail: benedikt.schulte@uni-wuerzburg.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_3

2 Motivation and Introduction to Service Differentiation

Companies increasingly rely on third parties for raw materials, intermediate products, finished products, and spare parts. While this approach allows them to streamline internal processes and usually to reduce costs, it also requires that they control the risk that suppliers will not be able to deliver requested items on time and in full. Therefore, companies monitor their suppliers' delivery performance, measuring them in terms of the service levels that suppliers commonly guarantee for their customers through contractually stipulated service-level agreements. For instance, [7] report that 70% of the retailers in the consumer goods industry monitor their suppliers' service levels.

All customers prefer high levels of service, but some customers value high levels of service more than others and will trade product characteristics like price for higher service levels. For instance, reliable product availability is important for customers, who incur high shortage costs if an order is not fulfilled promptly, so paying a higher price is reasonable. However, other customers have lower shortage costs and attribute less value to product availability and more to price. Additionally, a single customer's requirements in terms of service levels may vary between different orders—higher when the customer places an emergency order and lower for standard orders that are less time-sensitive.

An example of variations in the required service levels concerns the provisioning of spare aircraft parts. Because airlines require spare parts to be available worldwide within few hours, the provisioning of these parts is usually handled by third-party spare-parts providers. Here, a particular challenge is that the importance of product availability varies not only between different customers but also between order types. In particular, the service levels required for emergency orders (which are usually termed "aircraft on ground") are higher than those for standard replenishments or scheduled maintenance.

Samii et al. [2] present an example for the varying importance of product availability from another domain. They discuss the case of influenza vaccines, where higher levels of vaccine availability are required for critical population segments (e.g., healthcare professionals, elders, and, children) while lower levels are acceptable for the general public.

As a consequence of the varying importance of product availability, the customers of companies that offer a single guaranteed service level frequently push for higher levels of service even at higher prices, while other customers demand lower prices without caring for availability. Clearly, standard service-level guarantees with a single service level for all customers cannot content all customers. Service-level menus overcome the shortcomings of single service-level guarantees: the company posts several combinations of prices and service levels and allows customers to choose from the options according to their needs.

The introduction of service-level menus changes how companies interact with their customers by allowing customers to choose among several offerings to match their needs. However, introducing several service-level guarantees also requires that

the offering company change how it operates, as the following example illustrates: A technology company recognized the necessity of offering several service levels for a range of their products. However, its production and distribution network could not provide more than one level of service. As a result, the company offered different service levels and charged different prices, but all customers received the best possible service. Providing this high level of service to all customers led to higher-than-necessary costs for the company, and the policy carried the risk of upsetting customers who paid higher prices while receiving the same level of service as those who did not.

As this example shows, the use of service-level menus requires adapting the supply chain in order to provide differing levels of service. This can be achieved via inventory rationing, especially through critical-level policies that protect certain parts of the inventory for orders (resp. customers) that require a higher level of product availability. Critical-level policies function in an intuitive and easily implementable way. One can think of a two-bin inventory-management rule in which each item of stock on hand is kept in one of two physically or virtually separated locations (e.g., bins). All demands are filled from the first bin until it is exhausted, at which time high-priority orders are served from the second bin, while low-priority orders are rejected or backordered. Here, the critical level corresponds to the contents of the second bin. This analogy can be extended to more than two customer classes (bins).

Because the company chooses which service levels and which prices to offer and customers self-select from the various offerings of the service-level menu, the company's profits and the customers' level of satisfaction should both increase. However, a decision-maker who is considering introducing such a service-differentiation strategy must first answer several questions: First, the decision-maker should determine whether such a service-differentiation strategy is likely to significantly increase profits. Second, if this is the case, the decision-maker must determine the number of service levels to offer and the corresponding prices and service levels. Third, the decision-maker must determine the parameters of the corresponding critical-level policy.

Although inventory-rationing strategies in general and critical-level policies in particular have been well-studied (cf. [1] for research in a multi-period setting and [2] for research in a single-period setting), the existing research on service differentiation considers only the third of the three questions for decision-makers. The next section explains how to address the other two.

3 Summary of Findings

Because developing an integrated perspective on the supply and demand sides of service differentiation involves the study of various complex and interrelated problems, the presented research proceeds in three steps. The first step focusses on the operations side, studying how to manage multiple service levels and how additional customer classes (or offerings) affect the required inventory. The second step addresses

the relationship between the market side and the operations side by studying the joint optimization of price and inventory (without service differentiation). The third step concerns when the introduction of a service-level menu increases profits over that of a single undifferentiated offering and how to design optimal service-level menus.

To this end, one analytical setting is maintained throughout the discussion: A profit-maximizing, monopolistic firm supplies a single product from a single warehouse over a finite period of time to a set of heterogeneous customers. Prior to the selling period the firm purchases a number of units of the product. During the selling period, individual customer demands follow a Poisson process, as does the total of all customer demands because sums of Poisson processes are Poisson processes. Whether a given customer demand is fulfilled depends on the current pricing policy and the current inventory. Any remaining units of stock at the end of the selling period are either salvaged or held for future sale such that the company incurs either a salvage value or holding costs.

The first step (cf. [6]) addresses the question of inventory management. Assuming that a number of customers (or customer classes) and the corresponding demand rates and service-level guarantees are exogenously given, we develop an approach by which to determine the minimum required starting inventory and the corresponding critical levels and explore how the number of customer classes affects the required inventory. In order to determine the minimum required starting inventory, closed-form expressions for α and β service levels for an arbitrary number of customer classes and given system parameters are derived. As a byproduct, the derivation of the closed-form expressions characterizes the service levels in terms of when the critical levels are hit. Based on the service-level expressions and additional structural insights, we provide an algorithm with which to derive numerically the parameters of a critical-level policy (i.e., the minimum required starting inventory and the associated critical levels) using demand rates and service-level guarantees as input parameters. Schulte and Pibernik [6] also includes an extensive numerical study in which the system parameters, including the number of customer classes, vary.

The second step (cf. [4]) addresses the integration of pricing and inventory management without service differentiation, that is, the integrated optimization of price and inventory in a single-period make-to-stock or procure-to-stock setting with Poisson demand. In particular, I develop an analytical solution approach that covers a broad class of demand functions, including linear and iso-elastic demand, and explains how to use piece-wise linear approximations to handle the complex and/or discontinuous price-demand relationships that may occur in real-life situations.

Building on the aforementioned results, the third step (cf. [5]) addresses the question concerning how to design optimal service-level menus while considering the underlying inventory-management policy. Because such service-level menus allow the firm to price-differentiate based on its customers' service-level preferences, we term this service-level-based price differentiation "SLBPD". The contribution of our research is threefold:

- First, we provide an analytical formulation for the integrated optimization problem of designing a service-level menu and determining the corresponding parameters

of the underlying inventory-rationing policy, a problem that has not, to the best of our knowledge, been studied before.
- Second, our research reveals analytical and conceptional insights that are relevant beyond the scope of our research. In particular, we develop an equivalent problem formulation that links SLBPD and dynamic pricing, allowing us to use the rich body of research on dynamic pricing in order to gain a better understanding of SLBPD and helping to put service differentiation in perspective and to interpret our results.
- Third, building on these insights, we study when SLBPD is profitable and how best to design a service-level menu. In particular, our analytical and numerical insights show that, in many cases, service differentiation does not increase profits significantly.

One way to interpret this finding is that differentiating customers based on service levels alone is a weak differentiation lever only, that is, the price differences between offerings with differing service levels need to be small in order to prevent customers from switching to offerings with lower prices and service levels. Therefore, successful price differentiation requires service differentiation's being supported by presence of additional conditions or measures (e.g., pricing restrictions or further differentiation levers). Indeed, our research also shows that service differentiation can significantly increase profits if the company experiences pricing restrictions.

These results have immediate relevance for companies that consider to use SLBPD. In particular, decision makers from such companies learn that the potential profitability of SLBPD depends on the relationship between their current price and the optimal monopolistic price, the price-setting newsvendor price. If their current price is greater (or not significantly smaller) than the price-setting newsvendor price, then they should not pursue service-differentiation further. However, if (e.g., due to regulation, competition, customer expectations, or other influences) the current price is significantly lower than the optimal monopolistic price, then service differentiation has the potential to increase profits significantly.

Acknowledgements During the time of his dissertation, the author was supported by a fellowship granted by the Foundation of German Business (sdw).

References

1. Arslan, H., Graves, S.C., Roemer, T.A.: A single-product inventory model for multiple demand classes. Manage. Sci. **53**(9), 1486–1500 (2007)
2. Samii, A.-B., Pibernik, R., Yadav, P., Vereecke, A.: Reservation and allocation policies for influenza vaccines. Eur. J. Oper. Res. **222**(3), 495–507 (2012)
3. Schulte, B.: Integrated Segmentation of Supply and Demand with Service Differentiation. Ph.D. Thesis, U Würzburg (2015)
4. Schulte, B.: The Price-Setting Newsvendor with Poisson Demand. Working Paper (2016)
5. Schulte, B., Pibernik, R.: Profitability of Service-Level-Based Price Differentiation with Inventory Rationing. Production and Operations Management (forthcoming) (2016). doi:10.1111/poms.12677

6. Schulte, B., Pibernik, R.: Service differentiation in a single-period inventory model with numerous customer classes. OR Spectr. **38**, 921–948 (2016)
7. Thonemann, U., Behrenbeck, K., Küpper, J., Magnus, K.-H.: Supply Chain Excellence im Handel. Gabler, Wiesbaden (2005)

Part II
Master's Thesis Prizes

Improved Compact Models for the Resource-Constrained Project Scheduling Problem

Alexander Tesch

Abstract In this article, we study compact Mixed-Integer Programming (MIP) models for the Resource-Constrained Project Scheduling Problem (RCPSP). Compared to the classical time-indexed formulation, the size of compact models is strongly polynomial in the number of jobs. In addition to two compact models from the literature, we propose a new compact model. We can show that all three compact models are equivalent by successive linear transformations. For their LP-relaxations, however, we state a full inclusion hierarchy where our new model dominates the previous models in terms of polyhedral strength. Moreover, we reveal a polyhedral relationship to the common time-indexed model. Furthermore, a general class of valid cutting planes for the compact models is introduced and finally all models are evaluated by computational experiments.

1 Introduction

In the *Resource-Constrained Project Scheduling Problem* (RCPSP) we are given a set of n non-preemptive jobs $j \in \mathscr{J}$ with processing times $p_j > 0$ and a set of resources $k \in \mathscr{R}$ with capacity $R_k \geq 0$ where each job $j \in \mathscr{J}$ has a demand of $r_{jk} \geq 0$ units of resource $k \in \mathscr{R}$. Furthermore, there are precedence relations $\mathscr{P} \subset \mathscr{J} \times \mathscr{J}$ between the jobs where $(i, j) \in \mathscr{P}$ indicates that job i must end before job j starts. In the RCPSP we want to compute starting times for all jobs that satisfy the precedence constraints and such that at any point in time the resource consumptions of all active jobs does not exceed the capacities. The objective is to minimize the *makespan* which equals the total project duration.

The most common MIP formulation for the RCPSP is the time-indexed model of Pritsker et al. [10]. In this model, every job is assigned to a starting time within a discrete scheduling horizon. Many variants and extensions of the time-indexed model have been studied during the last decades, see for example [4, 5, 9]. But since the model size grows quadratically with the scheduling horizon, time-indexed

A. Tesch (✉)
Zuse Institute Berlin (ZIB), Takustraße 7, 14195 Berlin, Germany
e-mail: tesch@zib.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_4

models are still computationally intractable for large time horizons. This motivates the study of *compact* MIP models for the RCPSP whose size is strongly polynomial in the number of jobs. Currently, mainly two types of compact models are known. Artigues et al. [1] introduce a *flow-based* compact model where a resource flow determines the precedences between the jobs. Koné et al. [8] develop two *event-based* compact models that assign all jobs to a fixed position in the starting order of the jobs.

In this article, we introduce a new event-based compact model and we study the polyhedral relationship between our model, the two models of Koné et al. [8] and the time-indexed model of Pritsker et al. [10].

2 MIP Models

First, we briefly introduce the main modeling concepts of the time-indexed model of Pritsker et al. [10] and the two compact models of Koné et al. [8].

Time-Indexed Model (DDT) [5, 10]. The time-indexed case considers a discrete time horizon $\mathcal{T} = \{0, \dots, T\}$, discrete processing times $p_j \in \mathbb{Z}_{>0}$ and decision variables $x_{jt} \in \{0, 1\}$ that are one, if job $j \in \mathcal{J}$ starts at time $t \in \mathcal{T}$. Resource constraints are applied at every time point $t \in \mathcal{T}$, therefore the model size grows quadratically with T.

Event-Based Compact Models [8]. In the compact models of Koné et al., we are given a set of *events* $\mathcal{V}$ where an event denotes a time point where one or multiple jobs start. Every event $v \in \mathcal{V}$ is therefore correlated to a variable $t_v \geq 0$ that describes the start time of all jobs that start at event v. All events appear sequentially, that is $t_v \leq t_{v+1}$ holds for all $v \in \mathcal{V}$. Since n events are sufficient, the makespan is modeled by the variable $t_{n+1} \geq 0$.

(i) **On-/Off Event-Based Model (OOE)**. This model uses *activity variables* $u_{jv} \in \{0, 1\}$ that are one, if job $j \in \mathcal{J}$ is executed during the event interval $[t_v, t_{v+1})$ with $v \in \mathcal{V}$.

(ii) **Start-/End Event-Based Model (SEE)**. The model considers variables $y_{jv} \in \{0, 1\}$ and $\overline{y}_{jw} \in \{0, 1\}$ that are one, if job $j \in \mathcal{J}$ *starts* at event $v \in \mathcal{V}$ or *ends* at event $w \in \mathcal{V}' = \{v + 1 \mid v \in \mathcal{V}\}$ respectively.

In [12], stronger inequalities for OOE and SEE are introduced.

2.1 A New Compact Model

We now introduce a new event-based model, the *Disaggregated Position Model* (DP), which considers decision variables $z_{jvw} \in \{0, 1\}$ that are one, if job $j \in \mathcal{J}$ starts at event $v \in \mathcal{V}$ and ends at event $w \in \mathcal{V}'$. The model states

$$\min \quad t_{n+1} \tag{1}$$

$$\sum_{(v,w)\in\mathscr{A}} z_{jvw} = 1 \qquad \forall j \in \mathscr{J} \tag{2}$$

$$p_j \cdot \sum_{v\le v'<w'\le w} z_{jv'w'} \le t_w - t_v \qquad \forall j \in \mathscr{J}, (v,w) \in \mathscr{A} \tag{3}$$

$$\sum_{j\in\mathscr{J}} \sum_{v\le v'<w} r_{jk} \cdot z_{jvw} \le R_k \qquad \forall v' \in \mathscr{V}, k \in \mathscr{R} \tag{4}$$

$$\sum_{(v,w)\in\mathscr{A}: w\ge v'+1} z_{ivw} + \sum_{(v,w)\in\mathscr{A}: v\le v'} z_{jvw} \le 1 \qquad \forall (i,j) \in \mathscr{P}, v' \in \mathscr{V} \tag{5}$$

$$t_v \ge 0 \qquad \forall v \in \mathscr{V} \cup \{n+1\}$$

$$z_{jvw} \in \{0,1\} \qquad \forall j \in \mathscr{J}, (v,w) \in \mathscr{A}$$

where the objective (1) is to minimize the makespan. By inequalities (2) every job starts and ends at events $(v,w) \in \mathscr{A} = \{(v,w) \in \mathscr{V} \times \mathscr{V}' \mid v < w\}$. Inequalities (3) determine the time lag $t_w - t_v \ge 0$ between two events $(v,w) \in \mathscr{A}$ that is the maximum duration p_j of a job $j \in \mathscr{J}$ that is scheduled between events v and w. Inequalities (4) ensure that the resource consumptions of all active jobs at event $v' \in \mathscr{V}$ do not exceed the capacities. For each precedence relation $(i,j) \in \mathscr{P}$ inequalities (5) forbid that job j has a start event earlier than the end event of job i. In contrast to OOE and SEE, the DP model involves no big-M parameters or linearized expressions.

2.2 Linear Transformations

Between the compact models there hold the following linear transformations

$$u_{jv} = \sum_{v'\le v} y_{jv'} - \sum_{v'\le v} \bar{y}_{jv'} \qquad \forall j \in \mathscr{J}, v \in \mathscr{V} \tag{6}$$

$$y_{jv} = \sum_{v<w} z_{jvw} \qquad \forall j \in \mathscr{J}, v \in \mathscr{V} \tag{7}$$

$$\bar{y}_{jw} = \sum_{v<w} z_{jvw} \qquad \forall j \in \mathscr{J}, w \in \mathscr{V}' \tag{8}$$

$$u_{jv} = \sum_{v'\le v<w'} z_{jv'w'} \qquad \forall j \in \mathscr{J}, v \in \mathscr{V} \tag{9}$$

where (6) links OOE with SEE, (7)–(8) links SEE with DP, and (9) links OOE with DP by applying both transformations consecutively.

Denote by $\boldsymbol{\Phi}_1$ and $\boldsymbol{\Phi}_2$ the linear transformations (9) and (6) respectively. Furthermore, given a MIP model M, let P^M be the associated polyhedron of its LP-relaxation and let P_I^M be the integer hull of P^M, see [11].

Theorem 1 *For the linear transformations Φ_1 and Φ_2 it holds*

$$\Phi_1(P_I^{DP}) = \Phi_2(P_I^{SEE}) = P_I^{OOE} \quad \text{and} \quad \Phi_1(P^{DP}) \subset \Phi_2(P^{SEE}) \subset P^{OOE}.$$

Theorem 2 *Assume an instance of the DP model and expand it by setting $\mathscr{V} = \mathscr{T} = \{0, \dots, T\}$. Then take the restriction $z_{jvw} = 0$ for all $(v, w) \in \mathscr{A}$ with $w - v \neq p_j$ and project it to the z-variables z_{jvw} with $w - v = p_j$. The resulting model DP′ satisfies $P^{DP'} = P^{DDT}$.*

Theorem 1 states the equivalence between integer solutions of the models OOE, SEE and DP. According to their LP-relaxations, however, our new DP model dominates the compact models OOE and SEE of Koné et al. [8]. It further reveals that SEE is stronger than OOE.

Moreover, Theorem 2 states a relationship between the compact models and the time-indexed model DDT. In other words, DDT is obtained from DP by subsequent expansion, restriction and projection of the corresponding polyhedron P^{DP}. Complete proofs of Theorems 1 and 2 can be found in [12].

3 Primal-Dual Cutting Planes

Let $\mu_{jv} \geq 0$ be the duration of job $j \in \mathscr{J}$ in the event interval $[t_v, t_{v+1}]$ for an event $v \in \mathscr{V}$. We couple the μ_{jv} variables into the compact models by adding the inequalities

$$\sum_{v \in \mathscr{V}} \mu_{jv} \geq p_j \qquad \forall j \in \mathscr{J} \tag{10}$$

$$\mu_{jv} \leq p_j \cdot u_{jv} \qquad \forall j \in \mathscr{J}, v \in \mathscr{V} \tag{11}$$

which indicate that every job has its processing time distributed over the event intervals $v \in \mathscr{V}$ (10) but only at events where the job is active (11). Since the variables u_{jv} in inequality (11) belong only to OOE, the transformations (6) and (9) yield equivalent inequalities for SEE and DP.

A job subset $F \subseteq \mathscr{J}$ is called *feasible*, if all jobs in F can be scheduled in parallel. Let $\mathscr{F} \subseteq 2^{\mathscr{J}}$ denote the set of all feasible job subsets.

Theorem 3 *Given coefficients $\delta_j \geq 0$ for all jobs $j \in \mathscr{J}$ where $\sum_{j \in F} \delta_j \leq 1$ holds for all feasible subsets $F \in \mathscr{F}$ then the following inequalities are valid*

$$\sum_{j \in J} \delta_j \mu_{jv} \leq t_{v+1} - t_v \qquad \forall v \in \mathscr{V}. \tag{12}$$

Inequalities (12) yield strong valid cutting planes for the proposed MIP extension. They further generalize many valid inequalities that were originally proposed for an LP model of Carlier and Néron [3], for example: *energetic reasoning cuts, (lifted)*

cover cuts, clique cuts and *redundant function cuts*, see also [6]. As shown in [12], the cutting planes (12) originate from a primal-dual relation between the two linear programming models of Brucker and Knust [2] and Carlier and Néron [3]. Cutting planes of the form (12) can be computed by solving a linear program with generally exponentially many inequalities (for every $F \in \mathscr{F}$). Therefore, we generate only a constant number of such cuts in a preprocessing step.

4 Computational Results

Our models were tested on 480 instances of the PSPLIB [7] where each instance considers 30 jobs, 4 resources and various precedence graphs. Computations are performed on a 3.5 GHz Intel Xeon CPU, 16 GB RAM using CPLEX version 12.6. We implemented the time-indexed model DDT and the compact models OOE, SEE and DP. More specifically, we use a transformed but equivalent version of SEE [12] with a sparser constraint matrix what can be exploited by modern MIP solvers. All compact models include the extensions (10)–(11) for which we generated $n = 30$ primal-dual cuts (12) according to randomized objective coefficients. The time limit of each instance is 300 s. Our experimental results are illustrated in Table 1.

The columns *opt* and *ub* = *opt* show the number of instances where the optimal solution was found, provably and non-provably respectively. Moreover, the columns *#vars* and *#cons* represent the average number of variables and constraints in the models. Columns Δlb and Δub show the total difference of the computed lower- and upper bounds compared to the weakest model.

Among the compact models, the revised SEE model reveals the best performance mainly due to its sparse constraint matrix. The OOE model performs well in the primal but weak in the dual because it has a few number of binary variables but a weak LP-relaxation. Even though it constitutes the theoretically strongest compact model, DP shows weaker results than all other models. The main reasons are the huge number of binary variables and highly fractional LP-relaxations that consume a lot of computation time. More sophisticated preprocessing techniques might overcome this complexity in the future. In comparison to DDT, the compact models are slightly inferior on most instances. Interestingly, on a subset of generally hard instances the revised SEE model strictly outperforms DDT in the primal and dual because the

Table 1 Comparison of the MIP models

Model	Opt	ub = opt	#vars	#cons	Δlb	Δub
DDT	422	428	4980	9368	491	−377
OOE	265	392	900	9240	186	−600
SEE	295	407	1830	8123	415	−592
DP	219	324	13950	5370	0	0

extension cuts (10)–(12) yield drastically stronger lower bounds compared to DDT and to the original models of Koné et al. [8]. Even a small number of randomized cuts yields reasonably strong dual bounds [12].

Since the time horizon of any problem instance can be scaled arbitrarily, the compact models will always dominate DDT at a certain scaling factor.

References

1. Artigues, C., Michelon, P., Reusser, S.: Insertion techniques for static and dynamic resource-constrained project scheduling. Eur. J. Oper. Res. **149**(2), 249–267 (2003)
2. Brucker, P., Knust, S.: Lower bounds for resource-constrained project scheduling problems. Eur. J. Oper. Res. **149**(2), 302–313 (2003)
3. Carlier, J., Néron, E.: On linear lower bounds for the resource constrained project scheduling problem. Eur. J. Oper. Res. **149**(2), 314–324 (2003)
4. Cavalcante, C.C., De Souza, C.C., Savelsbergh, M.W., Wang, Y., Wolsey, L.A.: Scheduling projects with labor constraints. Discrete Appl. Math. **112**(1), 27–52 (2001)
5. Christofides, N., Alvarez-Valdés, R., Tamarit, J.M.: Project scheduling with resource constraints: a branch and bound approach. Eur. J. Oper. Res. **29**(3), 262–273 (1987)
6. Haouari, M., Kooli, A., Néron, E., Carlier, J.: A preemptive bound for the resource constrained project scheduling problem. J. Scheduling **17**(3), 237–248 (2014)
7. Kolisch, R., Sprecher, A.: PSPLIB-a project scheduling problem library: OR software-ORSEP operations research software exchange program. Eur. J. Oper. Res. **96**(1), 205–216 (1997)
8. Koné, O., Artigues, C., Lopez, P., Mongeau, M.: Event-based MILP models for resource-constrained project scheduling problems. Comput. Oper. Res. **38**(1), 3–13 (2011)
9. Möhring, R.H., Schulz, A.S., Stork, F., Uetz, M.: Solving project scheduling problems by minimum cut computations. Manage. Sci. **49**(3), 330–350 (2003)
10. Pritsker, A.A.B., Waiters, L.J., Wolfe, P.M.: Multiproject scheduling with limited resources: a zero-one programming approach. Manage. Sci. **16**(1), 93–108 (1969)
11. Schrijver, A.: Combinatorial Optimization, vol. A (2003)
12. Tesch, A.: Compact MIP models for the resource-constrained project scheduling problem. Master's Thesis, TU-Berlin (2015)

A Precious Mess: On the Scattered Storage Assignment Problem

Felix Weidinger

Abstract Induced by the rise of online retailing new storage strategies have evolved, designed to meet the demands of e-commerce warehousing. Although many of these new approaches have established over the last few years, literature on basic planning problems in these environments can be found only rarely. This paper points out the special needs of e-commerce warehousing and details the scattered storage strategy (also known as mixed-shelves storage) where unit loads are unbundled and single items are stored at multiple positions within the warehouse. This way, an item of an ordered product is always close by and the unproductive walking time of pickers is reduced. Based on the paper of Weidinger and Boysen (Scattered storage: how to distribute stock keeping units all around a mixed-shelves warehouse. Working Paper Friedrich-Schiller-University Jena, 2015) [8], the scattered storage assignment problem is presented and the processes in a scattered storage warehouse are described.

1 Introduction

E-commerce has gained a lot of importance in the recent years. However, consulting today's literature on warehouse management one has to observe that the revolution of e-commerce is rarely considered. Instead—referring to storage assignment strategies—only the two classical approaches *dedicated* and *shared storage* can be found in many cases (see, e.g., [1]). This dichotomy, though, seems to be incomplete nowadays, as the rise of e-commerce has led to an emergence of new storage concepts supporting the needs of e-commerce warehousing much better than classical approaches.

The most important challenges to be tackled managing the logistic processes of an online retailer can be summarized under the following four points [9].

F. Weidinger (✉)
Lehrstuhl für Operations Management, Friedrich-Schiller-Universität Jena,
Carl-Zeiß-Straße 3, 07743 Jena, Germany
e-mail: felix.weidinger@uni-jena.de
URL: http://www.om.uni-jena.de/

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_5

- *Small orders*: Typically, consumers order only small quantities of products. According to the personal information of an Amazon warehouse manager, the vast number of orders contains only one or two items.
- *Large assortment*: Pursuing the long tail strategy (see, e.g., [2, 5]), most online retailers have an assortment of goods much larger than classical retailers. The revenue of selling niche products often represents a significant part of the total revenue.
- *Scalability*: The amount of orders to be processed is highly volatile, since it varies on different levels. Some seasons are more high-selling than others and the same can be observed for differing days of the week.
- *Tight delivery schedules*: An important aspect of customer service is a fast delivery. This way, one of the most significant disadvantages of online shopping in comparison with retail shops, the lack of immediateness, is partly compensated. Most online retailers guarantee next day delivery and the percentage of shops offering same day delivery is growing. Amazons program *Prime Now* even provides a delivery within a 1-h timeframe after submitting the order. In consequence, all processes triggered by a customer order are highly time-critical.

One representative of the new generation of storage strategies tackling these challenges is the scattered storage strategy. Like most of its companions, scattered storage can rarely be found in today's literature although it is implemented in a vast number of warehouses worldwide, including basically all European Amazon warehouses. This paper, at first, details the basic idea of scattered storage as well as important processes in a scattered storage warehouse (Sect. 2.1). Afterwards, the scattered storage assignment problem by Weidinger and Boysen [8] is described (Sect. 2.2) and their most important findings on scattered storage warehouses are pointed out (Sect. 2.3). Finally, the paper is summarized (Sect. 3).

2 The Scattered Storage Strategy

The basic idea of the scattered storage strategy is to unbundle unit loads in the receiving area and store single items of one stock keeping unit (SKU) at differing positions within the warehouse, furnished with head-high shelves (low-level picking [4]). To use space more efficiently, items of different SKUs are assigned to the same storage bays of so-called mixed-shelves. Having multiple storage positions per SKU tends to always have an item of an ordered SKU close by, irrespective of the position within the warehouse. Based on this concept, highly performant and scalable picking processes especially suited for the needs of online retailers can be implemented. One possible implementation of the picking process in such a scattered storage warehouse is detailed in the following. The example is based on the description of processes in European Amazon warehouses by Weidinger and Boysen [8].

2.1 The Picking Process in a Scattered Storage Warehouse

To use the height of a building although having implemented a low-level picking strategy, Amazon warehouses are often constructed as multi-mezzanine systems. Each level is furnished with head-height shelves arranged in a rectangular layout. However, a picker, typically, remains on a single floor of the system during the picking process and changes stories only in exceptional cases. As the mean order volume is small, batching is implemented to utilize full picker capacity. Avoiding an additional sorting step subsequent to the picking process, each picker is equipped with a small maneuverable cart providing one standardized bin for each order currently picked.

Not having a structured and, consequently, learnable allocation of SKUs, pickers are highly reliant on an information system. Therefore, each picker carries a handheld scanner providing information about the next storage position to be visited as well as the name and quantity of the SKU to be picked. Guided by the device, the picker heads towards the storage position and, once arrived, scans the storage bay as well as the article to be picked before he/she adds it to the corresponding bin. The scanning process ensures that the right article is picked and, additionally, tags the storage position as available for restocking in the underlying warehouse management system.

Once an order is completely picked, the worker can hand off the bin containing all demanded items to a conveyor system, which transports it to the packing and shipping area of the warehouse. Entrances to this conveyor system are available at multiple positions spread all over the warehouse. At each station of this so-called *distributed depot system* the picker can obtain new empty bins to refresh the capacity of the picking cart, while new picking instructions are made available via the handheld scanner. In consequence, the picking tours are not bound to a central point of the warehouse anymore. Instead, the pickers roam the warehouse continuously, finalizing old and receiving new picking orders as well as empty bins at the distributed depot system. When processing heterogeneous orders of small volume, this strategy leads to a highly performant picking process.

Leaving behind the concept of a central depot, however, classical storage assignment strategies will fail, as they are grounded on a centralized layout. Weidinger and Boysen [8] identify this deficit in research and present a storage assignment strategy especially suited for scattered storage warehouses. Their approach is detailed in the next section.

2.2 The Scattered Storage Assignment Problem

The intention of the storage assignment problem is to support the picking process best. Therefore, classical approaches tend to assign SKUs with a high turnover rate near to the depot and SKUs demanded less frequently to more remote storage positions [1]. This way, SKUs to be picked more often can be found close to the start and end point of each tour, resulting in shorter picking tours on average. Planning

the assignment in a scattered storage warehouse, however, the initial position of the picker is not available, as it could be each of the numerous stations of the distributed depot system. Therefore, Weidinger and Boysen [8] suggest maximizing the scatter of items. This way, the average length to reach the nearest item of a given SKU is shortened, irrespective of the actual position of the picker.

To quantify the scatter of an assignment, they introduce so-called *measuring points*, which could be interpreted as possible start points of a tour. However, these measuring points are not necessarily located at depot stations, but rather at strategically chosen locations. The objective of the presented scattered storage assignment problem is to find a feasible assignment Γ such that the maximum distances to reach the nearest item of each SKU $i \in I$ starting from an arbitrary measuring point $\tau \in D$ are minimized. An assignment is feasible if none of the storage positions $s \in S$ is assigned more than once and all n_i items of each SKU $i \in I$ are allocated to compatible storage positions, given by the subsets $S_i \subseteq S$. Restrictions on storage positions ($\exists i \in I : S_i \neq S$) may be caused by product size or weight, for example.

To avoid shortages, not all storage positions have been emptied when planning a new assignment. Therefore, the distance to reach an item of SKU i starting from measuring point τ is bounded by the distances to the already assigned items. The bounds are given by problem parameters $\delta_{\tau i}$, while the shortest distances to newly assigned items are assumed to be returned by the function $\mathfrak{d}_{\tau i}(\Gamma)$ depending on the storage assignment Γ. Based on this notation, the objective value of an assignment Γ is calculated as given in formula 1. Note that Weidinger and Boysen [8] use a weighted sum of maximum distances. A higher weight w_i of SKU i might represent that it is more time-critical or more fast-moving than other SKUs and, therefore, has to be considered with a higher priority.

$$F(\Gamma) = \sum_{i \in I} w_i \cdot \max_{\tau \in D} \{\min\{\mathfrak{d}_{\tau i}(\Gamma); \delta_{\tau i}\}\} \tag{1}$$

Example: An instance of the storage assignment problem is depicted in Fig. 1. The warehouse has ten storage positions (see Fig. 1a). Four of them are open to be reassigned to two items of SKU 1 and one item of SKU 2 and 3, respectively. SKU

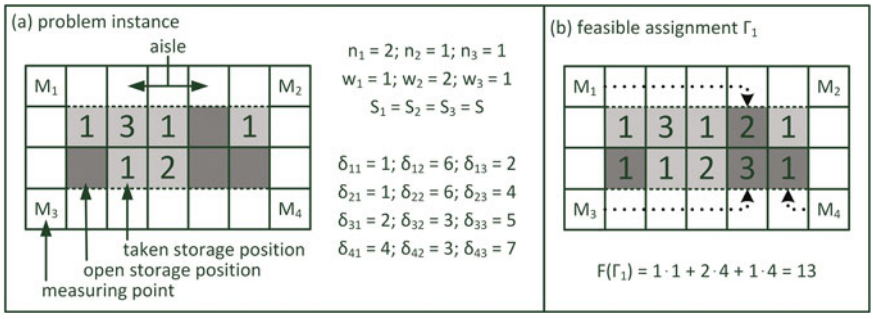


Fig. 1 Sketch of the example

2 is weighted by 2, while the remaining SKUs have a weighting factor of 1. As distances are measured by the number of squares to be crossed to reach the given storage position starting from the considered measuring point, the δ-values can be derived from the sketch of the warehouse. Note that the shelves are accessible only via non-cross aisles such that δ_{41} is 4 rather than 2. The feasible assignment Γ_1 (see Fig. 1b) improves the maximum distances given by the δ-values significantly. The objective value of this assignment amounts to $F(\Gamma_1) = 13$.

Weidinger and Boysen [8] formulate the optimization problem outlined above as a mixed integer model and prove NP-hardness in the strong sense. Additionally, they present suited heuristics to solve instances of real-world size in an adequate quality. Based on this, an extensive simulation study is realized. Its most interesting findings are summarized in the next section.

2.3 Effects of an Optimized Storage Assignment

Storage assignments are not optimized in most scattered storage warehouses. Instead, items are allocated to storage positions in a randomized manner. Weidinger and Boysen [8] show that picking tours are shortened by up to 20% on average and up to 50% in maximum when the storage assignment is planned more carefully using their optimization approach. Note that the picking tours are determined employing a modified nearest neighbor heuristic introduced by Daniels et al. [3]. While picker routing in a rectangular warehouse with single storage positions per SKU is a solvable case of the traveling salesman problem [6], picker routing in a rectangular scattered storage warehouse becomes NP-hard due to multiple available storage positions per SKU [7]. For this reason, a heuristic approach is used in the simulation study.

Furthermore, the replenishment level is identified as an important impact factor in the simulation study. Weidinger and Boysen [8] define it as the percentage of storage positions still occupied when replenishment takes place. Therefore, having a constant pick rate, a lower replenishment level leads to longer time intervals between two replenishment iterations. When using the randomizing assignment strategy, the picking process is supported best when replenishment is performed in short intervals. This way, the average quantity of storage positions per SKU is higher, resulting in shorter picking tours. Naturally, this effect is observable for optimized assignments as well. However, a second (antagonistic) effect is relevant for the optimizing strategy. As the storage assignment is manipulated by the picking process permanently, imbalances regarding the scatter may occur. These imbalances can better be fixed when the replenishment level is lower. This way, more open storage positions are available when planning the assignment and, consequently, a higher degree of scatter can be obtained, resulting in shorter picking tours on average, too. When using the optimizing approach, the replenishment level has to be selected such that both effects are properly traded off against each other, consequently.

Considering the higher effort for replenishment when using the optimizing strategy, Weidinger and Boysen [8] additionally study effects on the total effort including picking and replenishment. Even then the optimizing approach excels in scenarios where using scattered storage is expedient.

3 Conclusion

This paper treats the challenges in e-commerce warehousing, having led to a whole new generation of warehousing strategies. As a representative of these new approaches, the scattered storage strategy is detailed. With the help of the scattered storage assignment problem introduced by Weidinger and Boysen [8] the need for novel approaches for solving classical planning problems (e.g., storage assignment) in those new environments is demonstrated. The basic idea of the storage assignment strategy is outlined and the most important findings are summarized.

References

1. Bartholdi III, J.J., Hackman, S.T.: Warehouse & Distribution Science. Release 0.96. Supply Chain and Logistics Institute (2014)
2. Brynjolfsson, E., Hu, Y., Smith, M.D.: Consumer surplus in the digital economy: estimating the value of increased product variety at online booksellers. Manage. Sci. **49**, 1580–1596 (2003)
3. Daniels, R.L., Rummel, J.L., Schantz, R.: A model for warehouse order picking. Eur. J. Oper. Res. **105**, 1–17 (1998)
4. De Koster, R., Le-Duc, T., Roodbergen, K.J.: Design and control of warehouse order picking: a literature review. Eur. J. Oper. Res. **182**, 481–501 (2007)
5. Elberse, A.: Should you invest in the long tail? Harvard Bus. Rev. **86**, 88 (2008)
6. Ratliff, H.D., Rosenthal, A.S.: Order-picking in a rectangular warehouse: a solvable case of the traveling salesman problem. Oper. Res. **31**, 507–521 (1983)
7. Weidinger, F.: Picker routing in rectangular mixed shelves warehouses. Working Paper Friedrich-Schiller-University Jena (2016)
8. Weidinger, F., Boysen, N.: Scattered storage: how to distribute stock keeping units all around a mixed-shelves warehouse. Working Paper Friedrich-Schiller-University Jena (2015)
9. Weidinger, F., Boysen, N., Schneider, M.: Picker routing in the mixed-shelves warehouses of e-commerce retailers. Working Paper Friedrich-Schiller-University Jena (2017)

Integrated Location-Inventory Optimization in Spare Parts Networks

Patrick Zech

Abstract This research work is concerned with integrated location-inventory optimization in spare parts networks. A semi-Markov decision process (SMDP) is developed, formulated as linear program (LP) and finally, embedded into a set-covering problem framework. The resulting model is a mixed integer linear program (MILP) which integrates (1) strategic facility choice, (2) tactical base-stock level setting and (3) operational sourcing decisions. Due to the integration of these decision stages, physical and virtual inventory pooling opportunities can be evaluated at the same time. Experimental results emphasize the value of the integrated model compared to the sequential 'location first, inventory and sourcing second' approach. The cost savings are particularly high in networks with low fixed facility location cost, high shipment cost and high demand rates as virtual inventory sharing opportunities increase in these cases.

1 Introduction

After-sales service becomes increasingly important in today's marketplace as competition is strong and companies are looking for ways to distinguish themselves from their competitors. At the heart of after-sales service is providing the customer with spare parts in case of breakdowns that happen during regular operation. This work focuses on expensive and critical spare parts which are characterized by low demand rates and fast delivery requirements. The inventory holding cost of such parts are typically high which sets incentives to keep inventories low. Traditionally, low inventory levels have been achieved by consolidating multiple stocking points into one physical location and thereby, reducing the amount of system-wide safety stock [3]. However, the downside of this approach is that delivery times and outbound shipment cost typically increase since the centralized inventory is stored relatively far away from the markets. Instead of pooling inventory *physically*, there is also the possibility of

P. Zech (✉)
Logistics and Supply Chain Management, Technische Universität München,
Arcisstr. 21, 80333 Munich, Germany
e-mail: patrick.zech@tum.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_6

sharing inventory *virtually* among warehouses [5]. With this approach, the system-wide inventory level can be reduced as well while the distance between warehouses and markets tends to be shorter.

In this research work, we consider a spare parts manufacturer that outsources supply chain management to a third-party logistics service provider (3PL) and that needs to decide at which of the (already existing) warehouse locations to stock spare parts. This decision problem has the notion of the classical strategic facility *location* decision but, in fact, it is rather a facility *choice* or an assignment problem. For solving this decision problem, we propose a mixed integer linear program (MILP) that simultaneously evaluates physical and virtual sharing opportunities. Current research mostly focuses on physical pooling opportunities with the notable exception of Mak [4] who considers virtual inventory sharing in a location-inventory framework. To the best of our knowledge, there is no study yet that integrates both pooling variants in one model. The proposed MILP contains the following decision stages.

1. Strategic supply network design, i.e. at which warehouses to store spare parts.
2. Tactical inventory level optimization, i.e. which base-stock level to choose at each warehouse.
3. Operational sourcing, i.e. from which warehouse to satisfy spare part orders.

To evaluate virtual inventory sharing opportunities, it is necessary to include the inventory and sourcing decisions into the framework. The idea is that sourcing warehouses may vary dynamically depending e.g. on the current inventory level at each of the warehouses. Thus, demand can be allocated to multiple warehouses which then exhibit virtual inventory sharing. The MILP consists of a semi-Markov decision process (SMDP) that is formulated as linear program (LP) and embedded into a set-covering framework. The model is briefly presented in this article and our findings from an experimental study are provided. For further details on the model or the solution algorithm deployed, the reader is referred to [8].

2 Model Formulation

We consider a three-tiered supply chain consisting of one supplier, multiple warehouses $r \in R$ and markets $m \in M$. Each warehouse r replenishes items from an external supplier with infinite supply according to an $(S-1, S)$ review policy, i.e. the delivery of a part to a market immediately triggers a replenishment order at the respective sourcing warehouse r. Furthermore, the replenishment lead-time of warehouse r is exponentially distributed with mean $1/\mu_r$, where μ_r constitutes the replenishment rate of warehouse r per time unit. Assuming an exponential lead-time distribution appears rather restrictive at first glance—however, Alfredsson and Verrijdt [1] have shown that the overall system performance is rather insensitive with regard to the chosen lead-time distribution which makes our assumption robust. Each market m faces a Poisson demand process with an expected number of demand arrivals λ_m

per time unit. Furthermore, every market m can only be served by a subset of warehouses R_m because of service time constraints related to the geographical distance between warehouses and markets.

Cost of $t_{r,m}$ are incurred for shipping one item from warehouse r to market m. If no item is available at a warehouse within a market's service region, the part is express-shipped from an external supplier at cost of l_m. The unit replenishment cost of warehouse r are v_r and the unit inventory holding cost at warehouse r are h_r per time unit. Moreover, fixed cost of f_r are incurred if warehouse r is used to store spares.

For solving the outlined three-stage decision problem, we propose a MILP which integrates an SMDP with a classical set-covering model. The latter is concerned with the strategic network design decision and selects a subset out of a set of candidate warehouses. Inventory and sourcing decisions are modeled with an SMDP which is a reformulated version of the one in Seidscher and Minner [6]. The SMDP essentially models an inventory system that contains the candidate warehouses $r \in R$ as stocking points. By minimizing replenishment cost, inventory holding cost, shipment cost and express-shipment cost, the SMDP specifies in each state of the system from which warehouse to source an incoming part order. Thus, it determines the optimal sourcing policy for a given set of stocking points and base-stock levels.

The states $i \in I$ of the SMDP represent the allocation of inventory to the stocking points. We distinguish between auxiliary states $i \in I^A$ and decision states $i \in I^D$. The former is used to determine whether the next event will happen at a warehouse (arrival of an outstanding replenishment order) or at a market (arrival of a new spare part order). In those states, the system is not allowed to take a sourcing decision, i.e. to specify from which warehouse to source the demand of a market. In contrast, decision states $i \in I^D$ are concerned with taking these sourcing decisions $q \in R_{c_i}$ for a particular market $c_i \in M$.

Let us introduce the following sets and parameters. First, the sets $V^A(r,u)$ and $V^D(r,u)$ contain those states $i \in I^A$ and $i \in I^D$ that have an inventory level larger than u at warehouse $r \in R$, respectively. Second, $O(r)$ comprises those decision states $i \in I^D$ where warehouse $r \in R$ is out of stock. Furthermore, let U_r^{max} denote the (preprocessed) maximum possible base-stock level at warehouse r. U_r^{max} is not to be confused with a maximum storage capacity and is determined by optimizing an $M|M|S|S$ queue [8]. Moreover, the following decision variables are introduced.

- y_r Binary decision variable that indicates whether warehouse $r \in R$ is used for inventory placement.
- $S_{r,u}$ Binary decision variable that indicates whether the base-stock level u at warehouse $r \in R$ is active.
- $x_{i,q}$ Decision variable that denotes the long-run fraction of decision epochs where the system is in decision state $i \in I^D$ and decision $q \in R_{c_i}$ is taken.
- $x_{i,0}$ Decision variable that denotes the long-run fraction of decision epochs where the system is in auxiliary state $i \in I^A$.
- $z_{i,0,r,u}$ Decision variable that replaces the product $S_{r,u} \cdot x_{i,0}$, $\forall i \in I^A, \forall r \in R, u = 0, 1, \ldots, U_r^{max}$.

$$\min \quad \sum_{r \in R} f_r \cdot y_r + C(SMDP) \tag{1}$$

$$s.t. \quad \sum_{r \in R_m} y_r \geq 1 \qquad \forall m \in M \tag{2}$$

$$S_{r,u+1} \leq S_{r,u} \qquad \forall r \in R, u = 0, \ldots, U_r^{max} - 1 \tag{3}$$

$$S_{r,0} = 1 \qquad \forall r \in R \tag{4}$$

$$\sum_{u=1}^{U_r^{max}} S_{r,u} \leq U_r^{max} \cdot y_r \qquad \forall r \in R \tag{5}$$

$$\sum_{i \in V^A(r,u)} x_{i,0} + \sum_{i \in V^D(r,u)} \sum_{q \in R_{c_i}} x_{i,q} \leq S_{r,u+1} \qquad \forall r \in R, \forall u = 0, \ldots, U_r^{max} - 1 \tag{6}$$

$$\sum_{i \in O(r)} \sum_{q \in R_{c_i | q=r}} x_{i,q} \leq y_r \qquad \forall r \in R \tag{7}$$

$$z_{i,0,r,u} \leq S_{r,u} \qquad \forall i \in I^A, \forall r \in R, \forall u = 1, 2, \ldots, U_r^{max} \tag{8}$$

$$z_{i,0,r,u} \leq x_{i,0} \qquad \forall i \in I^A, \forall r \in R, \forall u = 1, 2, \ldots, U_r^{max} \tag{9}$$

$$z_{i,0,r,u} \geq x_{i,0} - \left(1 - S_{r,u}\right) \qquad \forall i \in I^A, \forall r \in R, \forall u = 1, 2, \ldots, U_r^{max} \tag{10}$$

$$y_r \in \{0, 1\} \qquad \forall r \in R \tag{11}$$

$$S_{r,u} \in \{0, 1\} \qquad \forall r \in R, \forall u = 0, 1, 2, \ldots, U_r^{max} \tag{12}$$

$$x_{i,0} \geq 0 \qquad \forall i \in I^A \tag{13}$$

$$x_{i,q} \geq 0 \qquad \forall i \in I^D, \forall q \in R_{c_i} \tag{14}$$

$$z_{i,0,r,u} \geq 0 \qquad \forall i \in I^A, \forall r \in R, \forall u = 1, 2, \ldots, U_r^{max} \tag{15}$$

$$+ \textit{SMDP constraints} \tag{16}$$

The objective function is given by (1) which consists of two cost terms. The first part refers to the costs associated with the strategic facility choice decision. The second term denotes the total SMDP costs which is the sum of inventory holding cost in auxiliary states as well as shipment, express-shipment and replenishment cost in decision states associated with sourcing decisions.

Constraint (2) ensures that at least one warehouse location that can serve market m within the required service time window is used to stock spares. Constraint (3) represents the incremental definition of the $S_{r,u}$ variables and ensures that base-stock level $u + 1$ can only be active if the predecessor base-stock level u is also active. Moreover, constraint (4) requires that base-stock level $u = 0$ is active at each warehouse $r \in R$. Furthermore, constraint (5) connects the inventory and location decision, i.e. only at the selected warehouses inventory can be placed.

Constraints (6) and (7) connect the set-covering problem framework with the SMDP model. Constraint (6) applies the following logic: Those states that would involve inventory levels higher than the base-stock levels need to be forbidden, i.e. the relative fraction of being in that state (taking any decision) have to be equal to zero. Additionally, constraint (7) ensures that demand can only be assigned to out-of-stock warehouses that are also open (incurring unfavorable express-shipment cost).

When integrating the SMDP with the set-covering framework, the model (at first) becomes non-linear as binary ($S_{r,u}$) and continuous decision variables ($x_{i,0}$) are multiplied with each other. We resolve the non-linearity by introducing a new set of continuous decision variables $z_{i,0,r,u}$ that replace the product term. Furthermore, we add constraints (8)–(10) to the model. This approach is consistent with the literature, see e.g. [2]. Moreover, constraints (11)–(15) define the variable domains.

For the sake of clarity, the SMDP constraints as well as the SMDP objective function are not formulated explicitly in this article. The interested reader is referred to [8] for a full exposition of the MILP, in particular the SMDP. Nevertheless, in order to give a notion of the SMDP model, we provide the general LP formulation that can be used to solve SMDPs [7]. τ_i is the average time of being in state $i \in I$ and $p_{i,j,q}$ denotes the transition probability from state $i \in I$ into state $j \in I$ under decision $q \in Q(i)$. $C_{i,q}$ denotes the cost in state $i \in I$ associated with decision $q \in Q(i)$.

$$\min \sum_{i \in I} \sum_{q \in Q(i)} C_{i,q} \cdot \frac{x_{i,q}}{\tau_i} \tag{17}$$

$$s.t. \quad \sum_{q \in Q(j)} \frac{x_{j,q}}{\tau_j} - \sum_{i \in I} \sum_{q \in Q(i)} p_{i,j,q} \cdot \frac{x_{i,q}}{\tau_i} = 0 \qquad \forall j \in I \tag{18}$$

$$\sum_{i \in I} \sum_{q \in Q(i)} x_{i,q} = 1 \tag{19}$$

$$x_{i,q} \geq 0 \qquad \forall i \in I, \forall q \in Q(i) \tag{20}$$

The objective function (17) minimizes the sum of the expected long-run average cost per time unit. Constraint (18) refers to a set of balance equations which ensure that for any state $j \in I$ the long-run average number of transitions *from* state j per time unit are equal to the long-run average number of transitions *into* state j per time unit. Moreover, the convexity constraint (19) forces the sum of all $x_{i,q}$ variables (over all states and decisions) to be equal to 1. Furthermore, (20) requires $x_{i,q}$ to be non-negative.

3 Findings and Conclusion

The integrated model is compared to the sequential 'location first, inventory and sourcing second' approach which essentially maximizes physical pooling opportunities. In a network with 3 warehouses and 6 markets (3×6), three model input

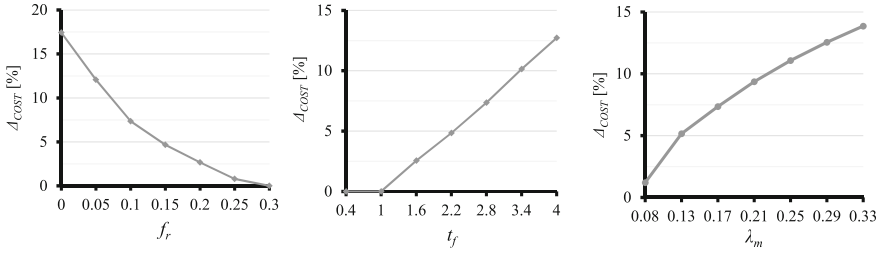


Fig. 1 Cost comparison between integrated and sequential approach in a 3×6 network

parameters are varied and the cost differences between integrated and sequential approach are measured. The experiments reveal that the cost savings (Δ_{COST}) are particularly high in networks with low fixed facility location cost (f_r), high shipment cost (t_f) and high demand rates (λ_m) as virtual inventory sharing opportunities increase in these cases, see Fig. 1. Note that t_f is a linear scaling factor for the shipment cost $t_{r,m}$.

Our results clearly show that there is a huge cost saving potential in evaluating both physical and virtual inventory sharing opportunities simultaneously rather than focusing on only one of the extremes.

Acknowledgements I would like to express my gratitude towards Prof. Dr. Stefan Minner whose comments and feedback significantly contributed to this work. Special thanks also go to Prof. Dr. Zuo-Jun (Max) Shen who hosted me as a visiting student researcher at University of California at Berkeley and also provided valuable input to my research.

References

1. Alfredsson, P., Verrijdt, J.: Modeling emergency supply flexibility in a two-echelon inventory system. Manage. Sci. **45**(10), 1416–1431 (1999)
2. Cui, T., Ouyang, Y., Shen, Z.-J.: Reliable facility location design under the risk of disruptions. Oper. Res. **58**(4-part-1) 998–1011 (2010)
3. Eppen, G.D.: Effects of centralization on expected costs in a multi-location newsboy problem. Manage. Sci. **25**, 498–501 (1979)
4. Mak, H.-Y.: Integrated supply chain design under uncertainty. Ph.D. Thesis, University of California at Berkeley (2009)
5. Paterson, C., Kiesmüller, G., Teunter, R., Glazebrook, K.: Inventory models with lateral transshipments: A review. Eur. J. Oper. Res. **210**(2), 125–136 (2011)
6. Seidscher, A., Minner, S.: A semi-markov decision problem for proactive and reactive transshipments between multiple warehouses. Eur. J. Oper. Res. **230**(1), 96–115 (2013)
7. Tijms, H.C.: A First Course in Stochastic Models. Wiley (2003)
8. Zech, P., Minner, S., Shen, Z-J.: Integrated location-inventory optimization in spare parts networks using Benders' decomposition. Working paper (2016)

Part III
Business Analytics and Forecasting

Towards Mathematical Programming Methods for Predicting User Mobility in Mobile Networks

Alberto Ceselli and Marco Premoli

Abstract Motivated by optimal orchestration of virtual machines in mobile cloud computing environments to support mobile users, we face the problem of retrieving user trajectories in urban areas, when only aggregate information on user connections and trajectory length distribution is given. We model such a problem as that of finding a suitable set of paths-over-time on a time-dependent graph, proposing extended mathematical programming formulations and column generation algorithms. We experiment on both real-world and synthetic datasets. Our approach proves to be accurate enough to faithfully estimate mobility on the synthetic datasets, and efficient enough to tackle real world instances.

1 Problem Statement and Modeling

Motivated by optimal orchestration of virtual machines in mobile cloud computing environments to support mobile users [1], we face the problem of retrieving user trajectories in urban areas. We partition the region covered by a mobile network into cells, one for each Access Point (AP), and we suppose to be given: (a) the adjacency matrix between cells, and (b) the demand in each cell at each point in time, that is the number of users connected to the corresponding AP. We also assume that an aggregated information about user mobility is given, namely the probability distribution of trajectory lengths. Our aim is to find an estimate on the trajectory, and more in general on the path of each user, in terms of sequence of cells traversed by the user during the considered time horizon. Since demand is usually easy to forecast, e.g. by time series analysis, our methods can be seen in the long term as means of *predicting* the corresponding user mobility. Our modeling approach (Sect. 1) is the following: first, we perform a time discretization and a trajectory length categorization. Then, we

A. Ceselli (✉) · M. Premoli
Dipartimento di Informatica, Università Degli Studi di Milano, via Bramante 65, Crema, Italy
e-mail: alberto.ceselli@unimi.it

M. Premoli
e-mail: marco.premoli@unimi.it

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_7

introduce extended mathematical programming formulations, inspired by flows over time models, having a polynomial number of constraints, but an exponential number of variables, and two hierarchical objectives. We devise column generation algorithms (Sect. 2): pricing problems are resource constrained minimum cost path problems, for which we provide ad-hoc dynamic programming procedures. We experiment on both synthetic datasets, obtained through generative models from the literature, and real world datasets from a major mobile carrier in Paris for which ground truth is not available (Sect. 3). Our approach proves to be accurate enough to faithfully estimate mobility on the synthetic datasets, and efficient enough to tackle real world instances. Our model is the following.

Data Let $T = \{1, \dots, |T|\}$ be a set of time slices and N be a set of APs, each lying at coordinates (x_i, y_i) in a plane that models our urban area. For each $t \in T$ and $i \in N$, let $d_i^t \in \mathbb{Z}^*$ be the number of users connected to AP i during time slice t. We denote as Ω the set of feasible *paths-over-time* (paths in the remainder), each being a sequence of APs whose cells are adjacent, and which are assumed to be visited by users in consecutive time slices. Notation-wise, for each $p \in \Omega$, we indicate with $p(t)$ the AP visited at time t in path p, and we suppose $p(t)$ to be set to a dummy value "–" if path p starts after, or ends before t. Let $l(p)$ be the total length of each path $p \in \Omega$, that is the sum of euclidean distances between consecutive APs in the path. The starting and ending APs of each path (the first and last values of $p(t)$ which are different from "–") identify a trajectory; the same trajectory can be identified by many feasible paths. Let $K = \{1, \dots, |K|\}$ be a set of classes, obtained by partitioning Ω according to the length of its paths. For each $k \in K$, let l_k (resp. l_{k-1}) be the upper (resp. lower) bound on the length of each path in class k, with $l_0 = 0$; let also $n_k \in \mathbb{Z}^*$ be the number of users whose path is in class k. From an application point of view, we assume (x_i, y_i) and d_i^t to be given, e.g. by a telecommunication operator, Ω to be easily definable, e.g. by Voronoi tessellations and street maps, and l_k and n_k to be estimated by previous knowledge on users travel distance distributions like [2].

Variables Our aim is to assess how many users are expected to follow a path over our time horizon, that we indicate as x_p for each $p \in \Omega$. We also consider the possibility that users enter or quit the system, or that simply data d_i^t is approximate, allowing a positive (resp. negative) correction $\bar{\varepsilon}_i^t$ (resp. $\underline{\varepsilon}_i^t$) for each $i \in N, t \in T$.

Constraints A feasible solution respects the following constraints:

$$d_i^t - d_i^{t-1} = \sum_{j \in N} \sum_{\substack{p \in \Omega \\ |p(t-1)=j \\ \wedge p(t)=i}} x_p - \sum_{j \in N} \sum_{\substack{p \in \Omega \\ |p(t)=j \\ \wedge p(t-1)=i}} x_p + \bar{\varepsilon}_i^t - \underline{\varepsilon}_i^t \qquad \begin{matrix} \forall i \in N, \\ \forall t \in T, t>1 \end{matrix} \qquad (1)$$

$$\sum_{\substack{p \in \Omega \\ |l(p)<l_k}} x_p \geq \sum_{k' \leq k} n_{k'} \qquad \forall k \in K \qquad (2)$$

$$x_p \geq 0, \bar{\varepsilon}_i^t \geq 0, \ \underline{\varepsilon}_i^t \geq 0 \qquad (3)$$

Constraints (1) resemble flow conservation, imposing the expected variation at time t in the number of users connected to AP $i \in N$ at time t to be consistent with the number of users arriving in i and those leaving i, potentially with corrections given by $\bar{\varepsilon}_i^t$ and $\underline{\varepsilon}_i^t$. We experimented on variants of (1), including a pure flow conservation formulation, without improvements. Constraints (2) imply that the number of users following a path in class k is at least the estimated one: cumulative values are used.

Objective We adopt a hierarchical bi-objective approach. Our primary objective is to find a setting of the variables explaining our data with minimum absolute value correction, that is we optimize the following linear program (LP):

$$\min\ \varepsilon = \sum_{t \in T} \sum_{i \in N} (\bar{\varepsilon}_i^t + \underline{\varepsilon}_i^t) \quad \text{s.t.} \quad (1), (2), (3)$$

Once an optimal ε value is found, as a secondary objective we try to match the path lengths distribution as close as possible; that is, we minimize the maximum difference between the number of users migrating on paths of each class k according to our solution, and the estimated one:

$$\begin{aligned} &\min\ \eta && (4)\\ &\text{s.t.} \sum_{\substack{p \in \Omega \\ |l(p) \in [l_{k-1}, l_k)}} x_p - n_k \leq \eta,\ \forall k \in K && (5)\\ &\sum_{i \in N} \sum_{t \in T} \bar{\varepsilon}_i^t + \underline{\varepsilon}_i^t \leq \varepsilon && (6)\\ &(1), (2), (3) \end{aligned}$$

2 Algorithms

Both problems are LPs. However, as the cardinality of Ω grows combinatorially, it is computationally infeasible to solve them directly. Instead we perform column generation on the set of variables x_p.

For the primary objective problem, let λ_i^t and μ_k be the dual variables associated to constraints (1) and (2), resp. The reduced cost of a variable x_p is

$$\bar{c}_p = - \sum_{t \in T\ |p(t) \neq '' - ''} (\lambda^t_{p(t-1)} - \lambda^t_{p(t)}) - \sum_{k \in K\ |l_k > l(p)} \mu_k.$$

For each $k \in K$, the search for the most negative reduced cost variable encoding a path in class k can be mapped into the problem of finding a minimum cost path in a time-expanded directed graph G = {N', A}, that has one node (i, t) for each pair of AP $i \in N$ and time slice $t \in T$, together with two additional dummy nodes acting as origin and destination; i.e. $N' = (N \times T) \cup \{(o, t_{-1}), (d, t_{T+1})\}$. The set A includes one arc $(i, t-1; j, t)$ connecting nodes $(i, t-1)$ and (j, t) if and only if the cells of

APs i and j are adjacent. The dummy origin (resp. destination) has an outgoing (resp. incoming) arc to (resp. from) every other node. Each arc $(i, t-1; j, t)$ has cost $w_{i,j}^{t-1,t} = \lambda_j^t - \lambda_i^t$ and length $l_{i,j}^{t-1,t} = \|(x_i, y_i) - (x_j, y_j)\|$, except those incident to either the origin $(o, 0)$ or the destination $(d, T+1)$, whose cost and length are set to 0. Indeed, the graph nodes are organized in layers, one for each time slice; paths in G can only be composed by nodes of different layers, and by arcs connecting one layer with the subsequent one. Modeling of waiting decisions is included, as represented by arcs $(i, t-1; i, t)$.

Not all paths are considered feasible for each class k, but only those starting from $(o, 0)$ and ending in $(d, T+1)$ whose sum of arc lengths falls into the range $[l_{k-1}, l_k)$. In principle, performing column generation means to solve a *resource constrained* minimum cost path problem for each $k \in K$. However, we propose an ad hoc dynamic programming algorithm, that optimize over all classes simultaneously, working as follows. We consider *labels* of the form $(C, L, (i, t))$, encoding partial paths starting from $(o, 0)$, ending in (i, t), whose sum of arc prizes and lengths are C and L resp. We *initialize* the algorithm, creating a single starting label $(-\sum_{k \in K} \mu_k, 0, (i, t))$ for each $i \in N, t \in T$; then, we proceed layer by layer and node by node, that is, for each $t \in T$ and for each $i \in N$, we iteratively *select* each label $(C, L, (i, t))$ and *extend* it to all the nodes $(j, t+1)$ having $(i, t; j, t+1) \in A$, creating a new label $(C', L', (j, t+1))$ for each of them that has $L' = L + l_{i,j}^{t,t+1}$ and

$$C' = C + w_{i,j}^{t,t'} + \sum_{k \in K \,|\, L < l_k \wedge L' \geq l_k} \mu_k.$$

The creation of labels having $L' \geq l_{|K|}$ is skipped, as encoding infeasible paths. After treating each label we *check dominance* rules: if any label $(C'', L'', (j, t+1))$ has already been created, having $C'' \leq C'$ and $L'' \leq L'$, at least one inequality being strict, then $(C', L', (j, t+1))$ is fathomed; similarly, if $C'' \geq C'$ and $L'' \geq L'$, at least one inequality being strict, then $(C'', L'', (j, t+1))$ is fathomed. We stop when all pairs (i, t) have been considered. All labels whose cost C is negative encode paths of negative reduced cost. We remark that, given the laminar structure of constraints (2), this aggregated dynamic programming algorithm is able to produce in a single run the labels of all non dominated paths for each class k; a formal proof is omitted. This allows us on one side to improve efficiency, since only one resource constrained minimum cost path problem needs to be solved at each column generation iteration, and on the other side to obtain an effective multiple pricing strategy, that consists in enlarging the set $\bar{\Omega}$ at each column generation iteration with the minimum reduced cost path for each class $k \in K$, if any of negative reduced cost exists.

The same algorithm is used for the secondary objective problem (1)–(6). Formally, since the structure of constraints (5) is not laminar anymore, the dominance rules need to be slightly relaxed to take into account of the contribution of the new dual variables. In our implementation, instead, we found it computationally useful to keep the original rules and resort to heuristic pricing. As discussed in Sect. 3 the routine obtained in this way proved to be able to produce high quality solutions with limited effort.

3 Dataset Generation and Experiments

Unfortunately, no ground truth is available on our real-world dataset. Therefore, in order to test both the computational viability and the prediction accuracy of our methods, we proceed as follows. First, we draw APs coordinates at random, and we generate instances as collections of user paths-over-time on this set of APs; we refer to such a collection as the *original* paths. Then the number of users d_i^t connected to each AP $i \in N$ at time $t \in T$, and the details l_k and n_k of path length classes $k \in K$, are computed and used as sole input of our methods. Therefore the full collection of *original* paths is kept only for cross-checking (as post-processing) the quality of *predicted* paths, that are those produced as output solutions of our methods.

We propose two generative models of original paths. The first is a simple *ad-hoc* model: given the number of users U as input, for each of them we create a path whose length is drawn from a power law distribution, and whose starting time is chosen uniformly at random. We assume that one hop is made in each time slice, in a graph having one node for each AP, and one edge between each pair of APs whose Voronoi cells are adjacent. The second is a *Point of Interest* (*POI*) generative model, reproducing the behavior of users during rush hours [2]: we randomly define a set $S \subseteq N$ of residential points and a set $D \subseteq N$ of destination POIs. We randomly draw the starting (resp. final) position of each user from bivariate normal distributions centered in a user residential point of S (resp. POI of D). Attractiveness of APs and transition probabilities are built following [2]. One path is finally generated for each user, choosing a residential point uniformly at random, a destination AP at random according to the transition probabilities, computing the shortest path in the adjacency graph described previously, assuming one hop for each time slice.

Our algorithms are implemented in C++ using CPLEX 12.6 as LP solver; the tests are performed on a PC with i7 4.0 GHz CPU and 32 GB RAM. For experiments we use a synthetic set of 300 APs with coordinates randomly drawn from a single bivariate normal distribution, and considering 15 time slices. These values match well those of real applications [1]. Given this fixed set of APs, we create 5 instances with $U = 40000$ for each generative model. Given the lengths of all paths in each instance, we compute 100 path length classes, with l_k values given by the percentiles of lengths distribution. We first assess the computational viability of our methods. Table 1 reports, for each stage of our algorithm (column blocks) and for each generative model (table rows), the avg. number of column generation iterations,

Table 1 Computational efficiency

Gnr. model	1st stage				2nd stage				Total t.
	CG iter	Master t.	Pricer t.	n. paths	CG iter	Master t.	Pricer t.	n. paths	
Ad-hoc	81.0	0.66	1.27	59.08	32.8	8.61	2.35	68.12	516.4
POI	121.4	1.96	1.75	57.89	19.8	26.94	2.82	60.45	1019.4

Table 2 Prediction accuracy

δ	1st stage					2nd stage				
	3%	5%	10%	20%	40%	3%	5%	10%	20%	40%
Ad-hoc	17.81%	24.99%	34.81%	41.19%	56.33%	24.91%	35.03%	57.54%	79.30%	97.55%
POI	4.67%	8.29%	21.77%	50.63%	78.64%	5.60%	9.91%	26.16%	61.11%	94.20%

the avg. execution time of each master LP optimization (in sec.), the avg. execution time of each dynamic programming pricing algorithm (in sec.), the avg. number of paths added at each column generation iteration; the total execution time (in sec.) is also reported. Values are averaged over the 5 instances of each generative model. Our methods show to be computationally stable, the most critical point being the master LP optimization during second stage optimization. Affordable computing times are also observed on a real-world dataset concerning about 600 APs in Paris [1].

Then we assess the accuracy of our methods in rebuilding mobility patterns from demand and path length distributions. Here we focus only in rebuilding user trajectories in terms of origin and destination, being the target of both the original application and related works in the literature [2]. We assume each prediction to be correct if both origin and destination APs of predicted paths fall within distance δ from origin and destination APs of original ones. We designed a maximum likelihood procedure, that is based on flow computations, and outputs the best matching between predicted and original paths. Let $\mathcal{N}(i,j)$ (resp. $\widetilde{\mathcal{N}}(i,j)$) be the number of users whose origin is i and destination is j in the original paths (resp. predicted paths according to such a maximum likelihood matching). As accuracy measure we consider $\sum_{i\in N, j\in N} \min(\mathcal{N}(i,j), \widetilde{\mathcal{N}}(i,j))/U$. Table 2 reports, for each stage of our algorithm (column blocks) and for each generative model (table rows), the average accuracy obtained when different δ correction levels are allowed; δ values are reported as percentage of the radius of the instance region. As expected, exploiting second stage optimization substantially improves accuracy. Values of δ as low as 10% are enough to make predictions on ad-hoc models reach 57.5% accuracy, and values of δ of 20% yield average prediction accuracy of almost 80%. POI models are harder to predict. Still 60% accuracy can be achieved when $\delta = 20\%$.

Acknowledgements The project has been partially funded by Regione Lombardia—Fondazione Cariplo, grant n. 2015-0717, project REDNEAT.

References

1. Ceselli, A., Premoli, M., Secci, S.: Cloudlet Network Design Optimization. In: Proceedings of 2015 IFIP Networking, Toulouse (2015)
2. Liang, X., Zhao, J., Dong, L., Xu, K.: Unraveling the origin of exponential law in intra-urban human mobility. Nature—Scientific Reports, vol. 3 (2013)

Statistics Instead of Stopover—Range Predictions for Electric Vehicles

Christian Kluge, Stefan Schuster and Diana Sellner

Abstract Electric vehicles (EVs) can play a central role in today's efforts to reduce CO_2 emission and slow down the climate change. Two of the most important reasons against purchase or use of an EV are its short range and long charging times. In the project "E-WALD—Elektromobilität Bayerischer Wald", we develop mathematical models to predict the range of EVs by estimating the electrical power consumption (EPC) along possible routes. Based on the EPC forecasts the range is calculated and visualized by a range polygon on a navigation map. The models are based on data that are constantly collected by cars within a commercial car fleet. The dataset is modelled with three methods: a linear model, an additive model and a fully nonparametric model. To fit the linear model, ordinary least squares (OLS) regression as well as linear median regression are applied. The other models are fitted by modern machine learning algorithms: the additive model is fitted by boosting algorithm and the fully nonparametric model is fitted by support vector regression (SVR). The models are compared by mean absolute error (MAE). Our research findings show that data preparation is more influential than the chosen model.

1 Introduction

The use of EVs can play a central role in today's efforts to reduce CO_2 emission and slow down the climate change [10]. Despite research funding and public support, consumers react cautiously to current offers of the EV market. Surveys show that

C. Kluge (✉) · S. Schuster · D. Sellner
Technische Hochschule Deggendorf, Edlmairstraße 6 und 8,
94469 Deggendorf, Germany
e-mail: christian.kluge@th-deg.de

S. Schuster
e-mail: stefan.schuster@th-deg.de

D. Sellner
e-mail: diana.sellner@th-deg.de

C. Kluge
Technologiecampus Grafenau, Hauptstraße 3, 94481 Grafenau, Germany

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_8

two of the most important reasons against the purchase or use of an EV are its short range and long charging times [13].

While the problem of long charging times is of technical nature, the problem of short range has also a psychological dimension known as range stress, the fear of running out of energy on an open road. Especially for new users in electric mobility this mental pressure is intensified by a highly unreliable range prediction offered by car itself. The built-in range prognosis of cars is often based on the EPC of the immediate past. Therefore, in mountainous regions, where elevation changes are frequent and high, the range prognosis varies drastically with the elevation profile of the passed route. To better support drivers, the project "E-WALD—Elektromobilität Bayerischer Wald" equips EVs with tablet computers that visualize the remaining range by a polygon drawn on navigation map.

One way to estimate the range of an EV is to predict the EPC along routes that may be travelled. In this study, we describe the development and comparison of different models to choose the best model for estimating the EPC. The considered models are a simple multivariate linear regression fitted by OLS, a linear median regression also known as least absolute deviation (LAD) regression fitted by quantile regression, an additive model fitted by a boosting algorithm and a fully nonparametric model fitted by a SVR. Our approach is driven by the goal to estimate EPC in a way that is as independent from car model specific properties as possible. This will allow to apply the modelling process to a wide variety of vehicles from different car manufacturers.

The structure of this paper is as follows: In Sect. 2 we describe how the data was obtained and prepared. Section 3 presents the process of model development. The model evaluation is given in Sect. 4, and Sect. 5 concludes this work with a short discussion.

2 Data Description and Preparation

Data were collected from Nissan LEAF vehicles that are part of a commercial car fleet operated by the E-WALD GmbH. To store the data, tablet computers which constantly record the car trips have been installed in these EVs.

The data, such as battery power, ambient temperature, speed, heater consumption, as well as GPS coordinates (latitude and longitude), were collected with an interval of 1 s during the trips from September 2014 to January 2015 for 7 Nissan LEAF vehicles. To improve the quality of the data base, erroneous data and outliers have been removed. The features of the data are as follows: length of trips is between 3 and 75 km, duration of trips is between 5 min and 1 h, temperature is between −4 and 25 °C. After filtering, about 385 trips can be used for further analysis.

Our approach is to estimate the EPC independent from specific car models. We therefore concentrate on external factors such as elevation difference and temperature, and investigate their influence on the EPC. To distinguish the influence of ascending versus descending slope on the EPC, we introduce the notion of positive elevation difference (PED) which is defined by the sum of meters a car travelled

through ascending slope and negative elevation difference (NED) which is correspondingly defined by descending slope. In this study, a trip is divided into parts of by exactly 3 km travelled distance. In order to estimate EPC in GID (a Nissan LEAF internal unit which amounts to 80 Wh) per 1 km and slope, the entries on EPC, PED and NED have to be divided by the respective distance travelled (distance-based dataset).

3 Model Development

In literature, there are a lot of different methods for fitting linear models. The most prominent method is OLS regression. Besides, least absolute deviation (LAD) regression is also often used. While OLS is based on estimating the mean of a distribution, LAD is based on estimating the median. The additive model is fitted by a boosting algorithm. The first boosting algorithm in machine learning was designed for binary classification [3, 4]. According to Friedman [5], boosting can be interpreted as a gradient descent algorithm in a function space. Bühlmann and Yu [2] introduced component-wise functional gradient descent boosting for additive models. An overview is given by [1]. The variant of boosting algorithm that was used is based on estimating the median. The fully nonparametric model is fitted by SVR. SVR is a generalization of support vector machine (SVM), which was originally designed for binary classification [11, 12, 14]. These methods belong to the wide class of methods which are based on penalized risk minimization and, therefore, are most suitable for fitting nonparametric models as they balance the trade-off between complexity and goodness of fit, c.f. [7, Chap. 5].

Model Assumptions. At first, the dataset of the recorded tracks is used for a descriptive analysis to reveal interdependencies and relevant variables that are useful predictors for the EPC. Possible variables are shown in Table 1. Therefore we selected PED and NED as important variables and assumed a linear influence on the EPC. So the following basic functional structure was chosen:

$$\frac{\text{EPC}}{\text{km}} = \beta_0 + \beta_1 \cdot \text{PED} + \beta_2 \cdot \text{NED} + \beta_3 \cdot \text{Temp}^2 + \beta_4 \cdot \text{Temp} \tag{1}$$

where $\beta_0, \ldots, \beta_4$ denote the parameters to be estimated.

Table 1 Correlation analysis on continuous data of Nissan LEAF, most relevant data are bold

Variable	PED/km	NED/km	Temperature	Mean velocity
r (EPC/km)	**0.4084**	**−0.4413**	−0.0446	0.0470

The Models. The dependent variable is EPC and independent variables are PED, NED, and temperature. Three models with different degrees of generality have been investigated. The simplest model is the linear model

$$y = \beta_0 + \beta_1 \cdot x_{pos} + \beta_2 \cdot x_{neg} + \beta_3 \cdot x_{temp}^2 + \beta_4 \cdot x_{temp} + \varepsilon \tag{2}$$

where y denotes the EPC, x_{pos} the PED, x_{neg} the NED, x_{temp} the temperature, ε the error term and β_i the parameter vector. A convenient generalization of a linear model is the additive model [6].

$$y = \beta_0 + f_{pos}(x_{pos}) + f_{neg}(x_{neg}) + f_{temp}(x_{temp}) + \varepsilon \ . \tag{3}$$

The difference to the linear model is that the additive model also captures nonlinear effects (f_{pos}, f_{neg} and f_{temp} are continuous functions). The study was done using the statistical software R where we applied the function gamboost with smooth P-spline base-learners PED, NED, and temperature [1, 8, 9]. Finally, we also considered the fully nonparametric model

$$y = f(x_{pos}, x_{neg}, x_{temp}) + \varepsilon \ . \tag{4}$$

As the additive model, the fully nonparametric model captures nonlinear effects. In contrast to the additive model, it also captures all kinds of interactions between independent variables so that the fully nonparametric model, in fact, is more general than the additive model. This was done using the R package e1071.

4 Results

As a measure for quality, the MAE has been chosen. Where n denotes the number of data points, y_i denotes the EPC (in GID) of data point number i and $\hat{y}_i$ contains corresponding estimate from the model, the MAE is given by

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} |y_i - \hat{y}_i| \ . \tag{5}$$

In case of more advanced nonlinear methods like Boosting and SVR, simply calculating MAE on the whole dataset is not appropriate; In order to avoid the problem of overfitting and to obtain honest values, the MAE was calculated using 10-fold cross-validation [7, Chap. 7]. Table 2 shows the results of the different models. All estimators which are calculated nearly have the same quality. The MAE of the LAD regression has the lowest value. Results were also compared with the global mean. It is simply the mean of the whole dataset. In doing so, the estimate $\hat{y}_i$ is always equal to the mean so that $\hat{y}_1 = \hat{y}_2 = \cdots = \hat{y}_n = \frac{1}{n} \sum_{i=1}^{n} y_i = \bar{y}$. The global mean acts

Table 2 Results of MAE for each model

Model	MAE	Improvement to global mean (%)	Improvement to OLS (%)
Global mean	1.098	0	−47.12
OLS	0.746	32.03	0
LAD	0.742	32.45	0.62
Boosting	0.744	32.25	0.33
SVR	0.743	32.37	0.51

as a benchmark because this is the result which could be obtained without collecting any data in the car. The 3rd and 4th column show the percentaged improvement to global mean and OLS respectively. Because all applied models have nearly the same performance, it is entirely sufficient to take the much simpler linear methods (OLS and LAD regression) for predicting the EPC.

5 Discussion

The perhaps most interesting aspect of the results is that the performance of models hardly makes a difference which estimator is chosen. During analysis it was also investigated how another data preparation will change the results. According to one possible way to prepare the data is to divide the trips into parts of 1 GID (of consumed energy) and to extrapolate the travelled distance to 1 km distance (energy-based data). So energy-based dataset and distance-based dataset (Sect. 2) in this study can be compared. As you see in Table 3 the estimated regression coefficients, the influence of independent variables are larger for the distance-based approach than for the energy-based approach. The MAE of the OLS with energy-based dataset was 0.886, very much higher than the MAE of OLS of the distance-based dataset (0.746, see Table 2). So the quality of estimators heavily depends on the way how the dataset is prepared but not which model is chosen. This is remarkable that the vast majority of research in data analysis is concerned with the choice of model and not with the topic of data preparation. In our case, the distance-based dataset is much smaller than the energy-based dataset ($n = 1476$ vs. $n = 4656$) but yields much better results. This

Table 3 Estimated regression coefficients (rounded)

Model	$\hat{\beta}_0$	$\hat{\beta}_{pos}$	$\hat{\beta}_{neg}$	$\hat{\beta}_{temp^2}$	$\hat{\beta}_{temp}$
OLS (energy-based data)	2.61	0.028	0.028	0.00014	−0.026
OLS (distance-based data)	2.64	0.067	0.041	0.00084	−0.061
LAD (distance-based data)	2.55	0.068	0.042	0.00064	−0.055

demonstrates, it is more important to have the right dataset, not the biggest dataset. In order to further improve quality of forecasts, it is interesting to investigate the history of forecasts separately for each trip. The current estimates are static. Therefore, it seems to be promising to improve estimations by adding dynamic and adaptive components.

Acknowledgements The E-WALD project and this study have been funded by the Bavarian State Ministry for Economic Affairs and Media, Energy and Technology.

References

1. Bühlmann, P., Hothorn, T.: Boosting algorithms: regularization, prediction and model fitting (with discussion). Stat. Sci. **22**, 477–505 (2007). doi:10.1214/07-STS242
2. Bühlmann, P., Yu, B.: Boosting with the l_2-loss: regression and classification. J. Am. Stat. Assoc. **98**(462), 324–338 (2003). doi:10.1198/016214503000125
3. Freund, Y., Schapire, R.E. (eds.): Experiments with a new boosting algorithm. In: Proceedings of the Thirteenth International Conference on Machine Learning Theory, Morgan Kaufmann Publishers Inc., San Francisco (1996)
4. Freund, Y., Schapire, R.E.: A decision—theoretic generalization of online learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997). doi:10.1006/jcss.1997.1504
5. Friedman, J.H.: Greedy function approximation: a gradient boosting machine. Ann. Stat. **29**(5), 1189–1232 (2001). doi:10.1214/aos/1013203451
6. Friedman, J.H., Stuetzle, W.: Projection pursuit regression. J. Am. Stat. Assoc. **76**(376), 817–823 (1981). doi:10.1080/01621459.1981.10477729
7. Hastie, T.J., Tibshirani, R.J., Friedman, J.H.: The elements of statistical learning: Data mining, inference, and prediction, 2nd edn., corr. at 7. printing edn. Springer Series in Statistics. Springer, New York (2001)
8. Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., Hofner, B.: Model-based boosting 2.0. J. Mach. Learn. Res. **11**, 2109–2113 (2010)
9. Hothorn, T., Bühlmann, P., Kneib, T., Schmid, M., Hofner, B.: mboost: Model-based boosting (2016). http://CRAN.R-project.org/package=mboost. R package version R package version 2.6-0
10. Li, C., Cao, Y., Zhang, M., Wang, J., Liu, J., Shi, H., Geng, Y.: Hidden benefits of electric vehicles for addressing climate change. Sci. Rep. **5**(9213) (2015). doi:10.1038/srep09213
11. Schölkopf, B., Smola, A.J.: Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond. MIT press, Massachusetts (2002)
12. Steinwart, I., Christmann, A., Jordan, M., Kleinberg, J., Schölkopf, B. (eds.): Support vector machines. Information Science and Statistics. Springer, New York (2008). doi:10.1007/978-0-387-77242-4. http://www.springerlink.com/content/uk1165
13. Türnau, M.: Befragung von elektrofahrzeug–mieterinnen. Mobilität, Gesellschaft und Technik (2014). http://digital.bib-bvb.de/webclient/DeliveryManager?pid=7179727Źcustom_att_2=simple_viewer
14. Vapnik, V.N.: Statistical Learning Theory. A Wiley-Interscience Publication. Wiley, New York (1998). http://www.loc.gov/catdir/description/wiley032/97037075.html

Improving the Forecasting Accuracy of 2-Step Segmentation Models

Friederike Paetz

Abstract The estimation of consumer preferences with choice-based conjoint (CBC) models is well-established. In this context, the use of Hierarchical Bayesian (HB) models, which estimate consumers' individual preferences is nowadays state-of-the-art. However, the knowledge of consumer preferences on a less disaggregated level, like segment-level, is key for demand predictions of non-customized products. Clustering individual HB data to achieve segment-level preferences is known as inappropriate, since 2-step segmentation approaches generally underlie 1-step approaches, e.g., Latent Class models. But, may the inclusion of different concomitant variables into the clustering process of individual CBC data relax that disadvantage? To answer this question, we used an empirical data set and compared the forecasting accuracy of 1- and 2-step approaches. While demographic variables showed small effects, psychographic variables turned out to heavily improve forecasting accuracy. In particular, 2-step approaches, that consider psychographic variables within the clustering process, showed a forecasting accuracy comparable to the one of 1-step approaches.

1 Motivation

The accommodation of consumer preferences is key for companies' survival. Hence, the estimation of consumer preferences and the derivation of consumer-oriented strategies constitutes a core research field for managers. During the last decades, the use of choice-based conjoint (CBC) methods has emerged as a valuable tool to assess consumer preferences by estimating part-worth utilities. Beside the assessment of consumer preferences, the consideration of preference heterogeneity is a must for successful decisions on optimal business planning. Within the context

F. Paetz (✉)
Clausthal University of Technology, Clausthal-Zellerfeld, Germany
e-mail: friederike.paetz@tu-clausthal.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_9

of CBC analysis, heterogeneity could be accounted for by different types of its representation, i.e., a discrete or a continuous representation. The assumption of a discrete distribution of consumer preferences results in segmentation models, which imply the estimation of segment-specific part-worth utilities. A continuous distribution of preferences results in Hierarchical Bayesian (HB) models and allows the derivation of individual parameters. The latter is nowadays state-of-the-art, since individual estimates constitute the most flexible form of heterogeneity's representation. Especially as an input for market simulations or for the derivation of optimal pricing strategies individual parameters are most relevant today.

Obviously, an individual customization of products as well as the determination and skimming of individual's willingness-to-pay may maximize company's revenues. However, with regard to the resulting cost, such a 1-to-1 marketing strategy seems not advisable at all. Rather, companies offer—if at all—a small number of product variations to accommodate preference heterogeneity. Hence, beside the need for individual estimates, the knowledge of a less disaggregated level of preference heterogeneity seems to be important. Segment-solutions, which allow for the derivation of segment-specific optimal product variations, may serve as a reasonable compromise between an individual and an aggregated level.

If individual parameter estimates are already observed, the trespass to segment-specific estimates by clustering methods, is straight forward. This 2-step segmentation approach, also known as *post-hoc segmentation* or *tandem approach* is heavily used in practice ([5], p. 32). However, it is well-known to underlie 1-step segmentation approaches, since the procedures in both steps of the 2-step segmentation approach optimize different criteria ([3], p. 374). Especially, forecasting accuracy of 2-step approaches may stay behind those of 1-step segmentation approaches. This behavior proves problematic, since correct demand predictions are most relevant for the derivation of product planning decisions etc. Hence, it is important to search for opportunities to improve the forecasting accuracy of 2-step approaches. This contribution aims to develop a method to improve the forecasting accuracy of 2-step segmentation approaches by incorporating concomitant variables, e.g., demographic or psychographic variables, into the clustering process of individual parameter estimates.

Within the next section, we provide a review of the theoretical construct of the Latent Class Multinomial Logit model as a surrogate for 1-step segmentation approaches. Furthermore, we introduce a HB model for the estimation of individual parameters, which are subsequently clustered into segments (2-step segmentation approach). In the third section, we use empirical data to compare the forecasting accuracy of both segmentation approaches and assess the appropriateness of incorporating different concomitant variables into the clustering process of a 2-step segmentation approach. Finally, we conclude our results by explicitly pointing out the appropriateness of different concomitant variables for improvements w.r.t. forecasting accuracy in 2-step segmentation approaches.

2 Latent Class and HB Models

In the context of CBC analysis, we assume a respondent to behave utility maximizing within a multi-alternative choice occasion. The utility of respondent j for a certain alternative m is contained in a latent unobservable variable U_{jm}, which satisfies

$$U_{jm} = x_m \cdot \beta_j + \varepsilon_{jm}.$$

Here, x_m describes the design vector of alternative m and β_j is the individual part-worth utility vector of respondent j. The random error term ε_{jm} includes all the effects, that are not contained in the deterministic part, but affect respondent's utility. If we assume the error term to be iid Gumbel distributed, the MNL model results and the probability of respondent j to choose alternative m could be given in closed form solution

$$P_{jm} = \frac{exp(\mu \cdot x_m \cdot \beta_j)}{\sum_{r=1}^{R} exp(\mu \cdot x_r \cdot \beta_j)}, \mu > 0, \tag{1}$$

where R describes the number of alternatives within a certain choice occasion and μ is a scale parameter.

The most popular 1-step segmentation approach, i.e., the Latent Class Multinomial Logit (LC-MNL) model, considers formula (1) and assumes the part-worth utility vector β_j to equal the segment-specific part-worth vector β_s, if respondent j is a member of segment s. To determine the segment-specific parameter estimates β_s as well as the relative segment masses, Maximum Likelihood estimation is performed (cp. [1]).

If we account for individual part-worth parameters in formula (1) and assume β_j to be Multivariate Gaussian distributed with mean σ and covariance matrix Σ, which in turn is assumed to be inverse Wishart distributed, the Hierarchical Bayesian Multinomial Logit (HB-MNL) model results. The estimation of a HB-MNL model constitutes in the estimation of σ and Σ as well as in the estimation of the conditional posteriori distribution of the individual parameters β_j. While the estimation of the mean and covariance matrix could be performed by Gibbs Sampling, the estimation of the conditional posteriori distribution of β_j is performed by a Metropolis-Hastings algorithm (cp. [4]).

Within the 2-step segmentation approaches, these individual part-worth utility estimates are subsequently clustered into segments with cluster analytic approaches to achieve segment-specific estimates. Within this clustering step, we are going to incorporate different concomitant variables.

3 Comparison of 1- and 2-Step Segmentation Approaches Based on Empirical Data

To compare the forecasting accuracy of 1- and 2-step segmentation approaches, we used an empirical data set in the product category of beer. In particular, we challenged 179 respondents with 15 choice occasions, which contained three beer alternatives (described by four attributes with two or three levels respectively) and a no-choice-option. In addition, the respondents conducted a (Big5-) personality test (see [2] for further information on the Big5 theory) and answered several socio-demographic issues, e.g., concerning gender, age (requested by four age classes) and size of household (requested by four classes, e.g., single-person household etc.).

For the 1-step segmentation approach, the data of 12 choice decisions by respondents served as input for the estimation of LC-MNL models with varying number of segments. We selected the 6-segment solution, since it provided the best trade-off between model fit and unique interpretability of segments. To measure forecasting accuracy, we calculated the first choice hit rate (*%1CH*) from the data of the remaining three (holdout) choice sets and achieved a hit rate of 63.69%.

For the 2-step segmentation approaches, we estimated the HB-MNL model (once more based on the data of 12 choice decisions by respondents) firstly. Subsequently, we clustered the individual estimates without (*CBC/HB*-clusters) and with (*CBC/HB+concomitant variable*-clusters) the consideration of concomitant variables. We selected the cluster-solution, that yielded the best/minimal value of Akaike's Information Criterion, respectively. Table 1 depicts the resulting number of clusters as well as the first choice hit rates of the CBC/HB cluster model and the CBC/HB+ cluster models. While the 6-cluster-solutions of the 2-step-CBC/HB and the -CBC/HB+ *personality* cluster model underlie the 6-segment-solution of the 1-step-LC-MNL model w.r.t. forecasting accuracy, all other 2-step-CBC/HB+ cluster models yield higher first choice hit rates. On first glance, this result contradicts general findings from literature, that 1-step segmentation approaches outperform 2-step approaches. However, the present behavior could be explained by the finer segmentation of the CBC/HB+ cluster models, which leads to a better accommodation of preference heterogeneity and in turn to a better forecasting accuracy. While the 7-cluster solution of the CBC/HB+*gender* cluster model exhibits a first choice hit rate of 63.87%, the 10-cluster-solution of the CBC/HB+*age class* model even shows a *%1CH*-value of 70.39%.

Table 1 Statistics for the CBC/HB and the CBC/HB+ cluster models

	CBC/HB	CBC/HB+ cluster models			
Conc. variables	–	Personality	Gender	Size of household	Age class
# of clusters	6	6	7	9	10
%1CH	61.64%	63.31%	63.87%	69.27%	70.39%

Table 2 Forecasting accuracy of 6-cluster-CBC/HB and -CBC/HB+ cluster models

	CBC/HB	CBC/HB+ cluster models			
Conc. variables	–	Personality	Gender	Size of household	Age class
$\%1CH$	61.64%	63.31%	62.94%	62.38%	61.86%

As nice as this high forecasting accuracy may be, the selected cluster-solutions partly lack unique interpretability of clusters and therefore constitute inappropriate for the derivation of managerial implications concerning product planning. This behavior of 2-step segmentation models is well-known (e.g., compare [3]) and comes not unexpected for our present data. The consideration of up to 10 clusters goes overboard for the considered product beer, which was described by four attributes only and therefore does not provide such as much space for heterogeneity.

Hence, in order to reduce the number of clusters and to provide a sound basis for the comparison of 2-step segmentation approaches with the LC-MNL model, we fixed the number of clusters to six, which equals the number of segments in the LC-MNL segment-solution. Table 2 depicts the associated first choice hit rates for the CBC/HB+ cluster models w.r.t. the 6-segment solutions. As expected, all 2-step-CBC/HB+ cluster models underlie the 1-step-LC-MNL model ($\%1CH = 63.69\%$) w.r.t. forecasting accuracy. However, while the pure clustering of individual part-worth utilities yields a first choice hit rate of 61.64%, which is 2.05% points below the hit rate of the LC-MNL model, forecasting accuracy increases, if concomitant variables are incorporated within the clustering process. While the consideration of socio-demographic variables leads to small improvements (0.22 (for age class) to 1.30% points (for gender)) in comparison to the pure clustering of individual part-worth estimates, personality as a concomitant variable yields the largest effect on forecasting accuracy. Furthermore, the first choice hit rate of the CBC/HB+personality model ($\%1CH = 63.31\%$) is on par with the hit rate of the LC-MNL model.

To gain further insight, we additionally used the individual background variables as segmentation bases and clustered the individual part-worth estimates once more. This constitutes another 2-step segmentation approach, but obviously does not consider consumer preferences as segmentation base. Table 3 yields the resulting first choice hit rates as well as the number of considered segments.

Table 3 Forecasting accuracy of segment models based on concomitant variables

Segmentation base	Personality	Gender	Size of household	Age class
# of clusters	8	2	4	4
$\%1CH$	48.04%	47.49%	49.35%	47.49%

While the number of socio-demographic segments arises from the mode of questioning, e.g., age was requested by four age classes, the number of psychographic segments is not fixed apriori. Therefore, we conducted a cluster analysis and selected the cluster-solution with a minimal Akaikes Information Criterion.

Forecasting accuracy of all segmentation models is far behind the predictive validity of the LC-MNL model ($\%1CH = 63.69\%$) and the CBC/HB cluster model ($\%1CH = 61.64\%$). Hence, preferences seem to be most appropriate as segmentation base w.r.t. the maximization of forecasting accuracy.

4 Conclusions

This study aimed to investigate, whether the incorporation of concomitant variables into the clustering process of 2-step segmentation approaches improves its forecasting accuracy. Therefore, we estimated individual part-worth parameters with a HB-MNL model and clustered those estimates without and with the consideration of several concomitant variables. Furthermore, we estimated a LC-MNL model as a surrogate for 1-step segmentation approaches. Under consideration of one empirical data set, we found, that the inclusion of concomitant variables within the clustering process of 2-step segmentation approaches pays off w.r.t. forecasting accuracy. While the incorporation of socio-demographic variables like gender, age and size of household leads to better forecasting accuracy than the pure clustering of part-worth utility estimates, the consideration of personality as a concomitant variable leads to a forecasting accuracy, which is on par with the one of the LC-MNL model. Hence, the incorporation of concomitant variables into the clustering process of a 2-step segmentation approach pays off w.r.t. forecasting accuracy, but, obviously, the degree of forecasting accuracy's improvement depends on the concomitant variable considered.

References

1. DeSarbo, W.S., Ramaswamy, V., Cohen, S.H.: Market segmentation with choice-based conjoint analysis. Mark. Lett. **6**(2), 137–147 (1995)
2. McCrae, R.R., Costa, P.T. Jr.: Five factor theory of personality. In: John, O.P., Robins, R.W., Pervin, L.A. (eds.) Handbook of Personality, 3rd edn. Guilford Press, Guilford (2008)
3. Ramaswamy, V., Cohen, S.H.: Latent class models for conjoint analysis. In: Gustafsson, A., Herrmann, A., Huber, F. (eds.) Conjoint Measurement and Applications, 3rd edn. Springer, Berlin (2013)
4. Train, K.: A Comparison of Hierarchical Bayes and Maximum Simulated Likelihood for Mixed Logit. Working Paper, University of California Berkley (2001)
5. Wedel, M., Kamakura, W.: Market Segmentation, 2nd edn. Kluwer Academic Publishers, Norwell (2000)

Field Service Technician Management 4.0

Michael Vössing and Johannes Kunze von Bischhoffshausen

Abstract Models for workforce planning and scheduling have been studied in operations research for decades. Driven by the Industrial Internet of Things new data sources have become available that have not yet been used to improve field service management. This paper proposes a research agenda towards leveraging this potential in the context of industrial maintenance. By combining predictive analytics (e.g. forecasting demand) with prescriptive analytics (e.g. determining optimal maintenance schedules) companies can decrease uncertainties in their maintenance planning, increase the availability of machines, decrease overall maintenance costs, and ultimately develop new business models.

1 Introduction

Many manufacturers of industrial machinery offer their customers supplementary repair and maintenance services which are provided by dedicated field service technicians. Providing these services economically requires efficiently managing a diverse workforce of highly specialized technicians. The widespread adoption of sensors and smart devices in the manufacturing industry—known as the Industrial Internet of Things or Industry 4.0—provides new opportunities to optimize established processes. By connecting machines and technicians in collaborative networks and leveraging the collected data, companies can manage uncertainties better and make more transparent decision [6].

Models for workforce planning and scheduling have been studied in operations research for decades. However current research has not yet incorporated the opportunities made possible by the Internet of Things. By combining previously unavailable data sources (e.g. collected by sensors in industrial machines) with emerging

M. Vössing (✉) · J. Kunze von Bischhoffshausen
Karlsruhe Service Research Institute (KSRI), Karlsruhe Institute of Technology (KIT),
Kaiserstraße 89, 76133 Karlsruhe, Germany
e-mail: michael.voessing@kit.edu

J. Kunze von Bischhoffshausen
e-mail: johannes.kunze@kit.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_10

technologies (e.g. analytic or data mining) innovative data-driven services can be developed and incorporated in traditional field service management solutions.

This paper is structured as follows. In Sect. 2 a short overview of asset management (see Sect. 2.1) and workforce management (see Sect. 2.2) fundamentals is given. In Sect. 3 an agenda for future research is proposed.

2 Fundamentals

The following section outlines the two main concepts field service technician management is build upon: (a) asset management and (b) workforce management.

2.1 Asset Management

Today many companies outsource the support of their infrastructure to external service providers. In the context of industrial asset management this is accompanied with a demand shift from the purchase of isolated maintenance and repair services to the purchase of long-term repair and maintenance contracts. For many companies outsourcing maintenance services is an attractive proposition as it allows them to (a) concentrate resources and investments on core competences, (b) focus on activities of strategic importance, and (c) minimize the economical risk associated with uncertain failure rates of machines and therefore uncertain demand for repair services over the lifetime of a specific machine. As a result, outsourcing maintenance, repair and overhaul has become a valid alternative to self-provisioning for many companies [2].

Manufacturers of industrial machinery have recognized that providing these services for their own customers can increase their revenue [4]. As a result companies that have traditionally focused primarily on building and selling the best machines, today offer supplementary services to complement their products. This trend—known as servitization—is defined by [1] as "innovation of an organizations capabilities and processes to better create mutual value through a shift from selling product to selling [product-service-systems]". Products (e.g. machines) are combined with auxiliary services (e.g. repair and maintenance activities) into integrated solutions. These systems are generally more distinctive, longer-lived, and easier to defend from competitors [1]. Finke and Hertz [4] have collected fundamental advantages for manufactures to offer these integrated solutions: (a) mean of differentiation to confront competition—especially given the fact that margins in product sales are constantly facing strong competition, (b) key interface to the customer for direct feedback, (c) continuous revenues and increased profitability through maintenance, repair and overhaul and (d) basis for data sharing and collection necessary for innovative business models [4]. These advantages illustrates the growing pressure to offer field services for machinery and equipment [5].

Offering maintenance, repair and overhaul services to a large number of customers requires developing effective field services networks. Managing these

networks and resources is a complex challenge due to (a) the constantly growing variety of products and parts that need to be managed simultaneously, (b) the requirement to respond quickly to uncertain demand, (c) the geographical discrepancy of demand and supply, and (d) the requirement to maintain a workforce that can support a highly heterogeneous (e.g. different technologies or systems) installed base [5]. To provision enough field service technicians to supports these networks—many machine manufactures have started looking for ways to optimize their field service networks.

In the context of industrial maintenance terminology often differs significantly between domains. Pintelon and Puyvelde [8] proposes a reasonable categorization: Asset maintenance is defined by three interconnected building blocks: (a) maintenance actions, (b) maintenance policies and (c) maintenance concepts. *Maintenance actions* are the basic interventions and tasks carried out by a technician, which can either be corrective (e.g. restoring a failed asset to an operational state) or preventive/precautionary (e.g. reducing the likelihood of failure of an asset). *Maintenance policies* on the other hand are defined as the mechanisms (e.g. sets of rules) that trigger the maintenance actions. Well known maintenance policies include (a) run-to-failure maintenance (also called breakdown maintenance), (b) time or usage based maintenance, (c) condition based maintenance, (d) opportunity based maintenance, and (e) design-out maintenance (e.g. redesign of parts that require high levels of maintenance). *Maintenance concepts* are high level combinations of maintenance actions and maintenance policies with suitable decision frameworks (e.g. objectives and strategies). Common concepts include (a) life-cycle costing, (b) total productive maintenance, and (c) reliability centered maintenance [8].

Research in the field of asset management is currently mainly focused on (a) predicting the failure of machines, (b) determining the correct maintenance concept, (c) optimizing the parameters of maintenance policies, or (d) offering support for outsourcing decisions [8].

2.2 Workforce Management

Efficient workforce management is one of the success factors of field service management. It requires companies to balance operational planning (e.g. scheduling or resource capacity planning, demand/supply matching) with strategical decision-making (e.g. skill demand forecasting, strategic planning or talent optimization) [7].

Operational Planning. Operational planning is the "feasible, efficient and effective planning of maintenance jobs" to coordinate technicians. It generally requires (a) evaluating incoming jobs (e.g. for missing information), (b) sequencing and scheduling and (c) allocating resources [8]. The last two steps have been studied in operations research for decades and have largely been influenced by supply chain management [7]. Today, operational planning is largely focused on technician scheduling. The main challenge is simultaneously managing (a) obligations from maintenance contracts as well as (b) temporal-uncertain demand for repair and maintenance services [3]. Efficient technician scheduling needs to take into account dif-

ferent types of jobs (e.g. repair, maintenance, installation), different technician skills, time windows (e.g. where jobs have to be started), different locations, as well as prioritization of jobs. The high complexity of scheduling problems makes finding optimal solutions (within reasonable time) difficult or even impossible. Solutions generally rely on (a) mathematical programming, (b) heuristics, and (c) empirical procedures. But even though a variety of approaches are available, due to the temporal-uncertain nature of repair and maintenance demand most companies still rely on operators to manually make these complex operational decisions [8].

Strategical Decision-making. As many companies adopt a service-centered mindset, talent is becoming a main competitive differentiator and needs to be managed accordingly. Today companies are not only competing for product superiority, but also for the human talent required to service their products. As field service technicians are usually specialized in varying technical disciplines (e.g. mechanics, electronics, automation) and given the fact that the overall complexity of industrial maintenance is rapidly evolving, skill management is essential for strategic technician workforce management [8]. This shift requires planners and managers to not only focus on operational challenges but also on the human aspect and the complex relationships present in a modern workplace. Strategic workforce management focuses on fostering collaboration, cross-training employees, providing attractive career environments, learning response curves, burnout, accelerations/slowdowns, sensitivity towards fairness in workload, and absenteeism [7]. This requires not only classical time, attendance and absence management, but also advanced worker tracking, demand and supply forecasting, scheduling and optimization and employee participation [7].

3 Towards Field Service Technician Management 4.0

As outlined in the previous section managing field service technician requires decision supports systems for (a) operational planning and (b) strategic decision-making [5].

In the Industrial Internet of Things a variety of data is captured from assets that need to be repaired, maintained, overhauled or installed. At the same time improved data mining algorithms have made previously underutilized data sources—which often contain large amounts of unstructured data—utilizable. Leveraging these data source for predictive maintenance (e.g. forecast maintenance demand of assets) has been one of the main use-cases. Unfortunately in closely related fields these advancements have largely been ignored. Little research has been conducted on how these data sources can be leveraged for field service management. Figure 1 provides an overview of how predictive and prescriptive analytics are interconnected with operational planning and strategic decision making.

Analytic models can support field service management in multiple areas: Predicting when a machine is likely going to fail and estimating in which cases preventive maintenance is feasible, transforms unplanned repair services which are difficult to

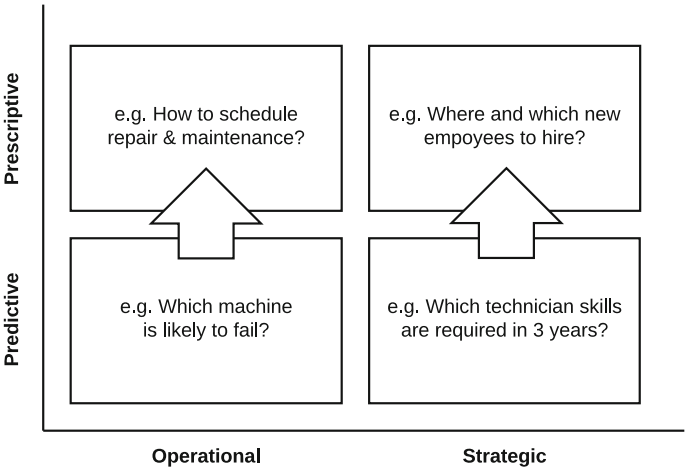


Fig. 1 A research agenda towards field service management 4.0 leveraging advanced analytic and the Industrial Internet of Things

manage into more manageable planned maintenance services. Predicting the duration of a specific service assignment reduces the amount of rescheduling in daily operations. Estimating service costs and the economical impact of unfulfilled service for customers enables companies to offer economical feasible full-service contracts. Traditional optimization models can leverage these predictions to optimize short- and long term technician schedules. On a strategic level forecasting which technician qualifications will be required in the following years can simplify capacity management by cross-training or hiring technicians ahead of time.

We have identified several aspects that will improve field service technician management in the future. Extending traditional operation research models (e.g. determining optimal jobs sequences and technician schedules) with advanced data mining techniques will fundamentally change traditional maintenance policies, service business models, and the underlying service network. In the long run industrial companies will increase the availability of their customers assets, decrease maintenance costs, and offer new business models (e.g. full-service contracts).

References

1. Baines, T.S., Lightfoot, H.W., Benedettini, O., Kay, J.M.: The servitization of manufacturing. J. Manufact. Technol. Manag. **20**(5), 547–567 (2009)
2. Campbell, J.D.: Outsourceing in maintenance: a valid alternative to self-provision. J. Qual. Maint. Eng. **1**(3), 18–24 (1995)
3. Ernst, A., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: a review of applications, methods and models. Eur. J. Oper. Res. **153**(1), 3–27 (2004)

4. Finke, G.R., Hertz, P.: Uncertainties in after-sales field service networks. In: The 2nd International Research Symposium in Service Management, Yogyakarta, Indonesia, 26–30 July 2011, pp. 186–194 (2011)
5. Hertz, P., Cavalieri, S., Finke, G.R., Duchi, A., Schönsleben, P.: A simulation-based decision support system for industrial field service network planning. Simulation2 **90**(1), 69–84 (2014)
6. Lee, J., Kao, H.A., Yang, S.: Service innovation and smart analytics for Industry 4.0 and big data environment. Proc. CIRP **16**, 3–8 (2014)
7. Mojsilović, A., Connors, D.: Workforce analytics for the services economy. In: Maglio, P.P., Kieliszewski, A.C., Spohrer, C.J. (eds.) Handbook of Service Science, pp. 437–460. Springer, Boston (2010)
8. Pintelon, L., Van Puyvelde, F.: Asset Management: The Maintenance Perspective. Acco (2013)

Part IV
Decision Theory and Multiple Criteria Decision Making

Optimal Placement of Weather Radars Network as a Multi-objectives Problem

Redouane Boudjemaa

Abstract This work proposes an approach to the optimal placement of a weather radar network based on solutions to a multi-objective optimization problem. Given a finite number of weather radars, a network is produced by taking into account the maximization of network coverage area and the minimization of network general cost. Several constraints on the solutions are considered such as terrain blockage, radar beam elevation and distance from power grid and roads. By transforming the search space into a gridded system, a reduction in the number of possible combinations of radar networks is achieved making the problem manageable in size. The multiobjective optimization problem is solved by four different evolutionary algorithms and the obtained results are analysed using different performance metrics. The proposed approach can serve as an analysis tool for a decision support system by providing meteorologists a set of Pareto-optimal solutions to assist in the selection of future prime sites for the installation of weather radars.

1 Introduction

Weather Radar Networks (WRN) have been initially used by meteorologists in studying severe weather phenomenons and the issuing of important and essential weather bulletins and information to all major agencies such as civil and military aviation, oil and gas companies, and civil defence. WRN have been commonly used in both the prevision and research of weather systems. The Next Generation Weather Radar (NEXRAD) system [9] for example has been efficiently used in the prediction, study and research of severe weather systems such as supercells, mesocyclones, tornado vortices, and various types of precipitation.

A difficult task in constructing these networks is determining adequate sitting sites of radars in order to meet certain conditions. A clear propagation of the radar beam for an altitude below one kilometre without being obstructed by terrain features is

R. Boudjemaa (✉)
Department of Mathematics, Faculty of Sciences, University M'Hamed Bougara of Boumerdes, 35000 Boumerdes, Algeria
e-mail: rboudjemaa@univ-boumerdes.dz

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_11

of extreme importance as the core of heavier precipitation lies within a hight above ground of 1000 m as pointed out by [10].

A mathematical model of the problem was achieved by [5] by establishing a well defined optimization problem. A recent work in determining the placement of WRs is investigated by [4]. Through the utilization of a genetic algorithm (GA) a maximization of the coverage area within a set of physical boundary condition is achieved.

2 Multiobjective Evolutionary Algorithms

Multiobjective Evolutionary Algorithms (MOEAs) are methods which approximate the Pareto Front (PF) by mimicking processes found in biological evolution. Hence, their aim is to find solutions that converge as close as possible to the true optimal solutions obtained so far during optimization. In the following paragraph, we mention some details about the MOEAs selected for the resolution of our below-mentioned problem. MOPSO algorithm [1] starts by generating a swarm with N random particles along with a set of leaders representing the nondominated particles. Position and velocity of each particle in the swarm is initialized and the fitness of each particle is evaluated. NSGA-II [2] computes a crowding distance for each individual by measuring the distance to its neighbouring individuals along each objective function dimension. The obtained crowding distance is then used to modify the fitness of each individual. The algorithm SPEA2 [12] uses an external archive A containing the nondominated solutions found so far. A strength value is assigned to both individuals in the archive and in the population. The MOGWO is an algorithm proposed by [6] in which the social and hunting technique of grey wolves are mimicked.

2.1 Performance Metrics

As a Pareto noncompliant metric, the Nondominated Vector Generation (ONVG) [8] measures the number of elements in a nondominated solutions set obtained by MOEA generation. Hence, a solution set with a large *ONVG* is preferred. The spacing (S) [7] is Pareto noncompliant metric which measures the minimum value of the sum of distances between consecutive solutions in a nondominated set. Zitzler and Thiele [11] proposed the performance metric *dominated hypervolume* (HV) as the union of hypercubes constructed using a reference point R, which can be taken as the vector of worst objective function values and a solution i of PF_{known} as the diagonal corners of the hypercube.

3 Problem Formulation

The latitudinal and longitudinal co-ordinates of the radars $(\phi_1, \lambda_1), (\phi_2, \lambda_2), \ldots$ are considered as design variables which are to be optimized. The following two objectives are considered.

Terrain Coverage In our work, a modified explicit enumeration method is used similar to the one used in [4]. The selected geographical region is discretized into a grid with a resolution of approximatively 0.09° (1 km) M latitudinal and N longitudinal spacing stored in a matrix $\mathbf{A}_{M\times N}$. We incorporate a new factor to our model using global digital elevation data at a resolution of 30 arc seconds (≈1 km) provided by the United States Geological Survey. The radar propagated beam is checked for terrain blockage at each grid point that either represent a potential radar site or is included inside the theoretical coverage layer of a radar [3] through the 4/3 law:

$$h = \sqrt{r^2 + R_e^2 + 2rR_e \sin\theta_e} - R_e \tag{1}$$

where h is the height of beam in km, r is the range of beam in km, θ_e is the elevation angle, and R_e is the effective earth's radius in km (4/3 the earth's radius). Using a binary encoding, all grid points are set initially to zero. The radar site along with the points which hight are below the radar beam and their slant range from the radar site is less than the maximum beam range are all set to one. The coverage area of a radar is the sum of all values of the grid points,

$$C_r = \sum_{i=1}^{M}\sum_{j=1}^{N} a_{i,j} \tag{2}$$

The minimization problem is then formulated with respect to (2) as

$$f_1 = 1 - \frac{\sum_{r=1}^{R} C_r}{T} \tag{3}$$

where R is the number of radars in the network and $T = \sum_{i=1}^{M}\sum_{j=1}^{N} 1$, is the total area of the studied region.

Network cost The economic and maintenance cost of installing a WR in R different sites is given by:

$$f_2 = \sum_{i=1}^{R} C_i x_i \tag{4}$$

where $C_i = q_1 EC_i + q_2 MC_i$ and $q_{1,2} \in [0, 1]$, $q_1 + q_2 = 1$ are weighting parameters. The parameter EC_i is the minimum economic cost of installing a WR in site i which

depends on the infrastructure and power availability. MC_i is the minimum maintenance cost parameter related to the distance of a site i to the nearest road accessible to truck traffic. Both EC_i and MC_i are obtained as the minimum *haversine* distance between the radar site and the nearest power line for the economic cost and road for the maintenance cost.

$$\Xi_i^k = \min_{\chi_j^k \in \Omega^k} \{2R_e \arcsin\left(\sqrt{\sin^2(\Delta\phi) + \cos(\phi_i)\cos(\phi_j)\ \sin^2(\Delta\lambda)}\right)\},\ k = a, b \tag{5}$$

where $\Delta\phi = \frac{\phi_i - \phi_j}{2}$, $\Delta\lambda = \frac{\lambda_i - \lambda_j}{2}$, (ϕ_i, λ_i) are the latitude and longitude coordinates of the radar site, and (ϕ_j, λ_j) are the latitude and longitude coordinates of a location $\chi_j^k \in \Omega^k$. The formula in (5) was used for both the economical cost, with $k = a$ and Ω^a being the power grid and for the maintenance cost with $k = b$ and Ω^b representing the road network.

4 Numerical Results and Discussion

The selected geographical region is the north of Algeria bounded by parallels 34° N and 36° N and meridians 3° E and 6° E with a total surface area of $6.076 \times 10^4 \text{km}^2$. The area is a mix of flat and complex surfaces supporting a diverse testing of the presented strategy. The analysis was conducted with a 1.1° radar beam elevation angle and the tower height of the radar is set to 15 m in order to reduce the effect of ground clutter. A theoretical coverage range of the radars is set to 45 km. For all the results presented in this section, the number of radars was limited to 5. Figure 1

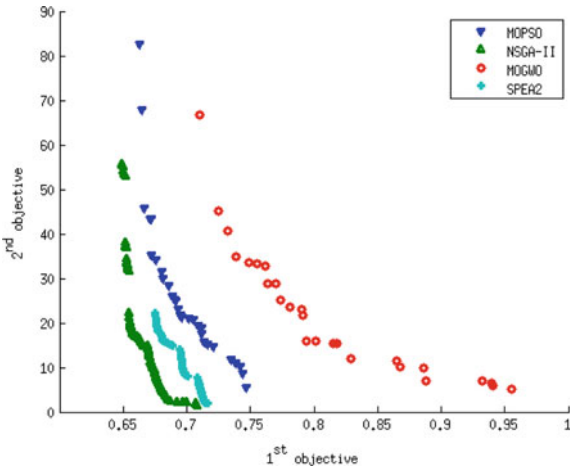
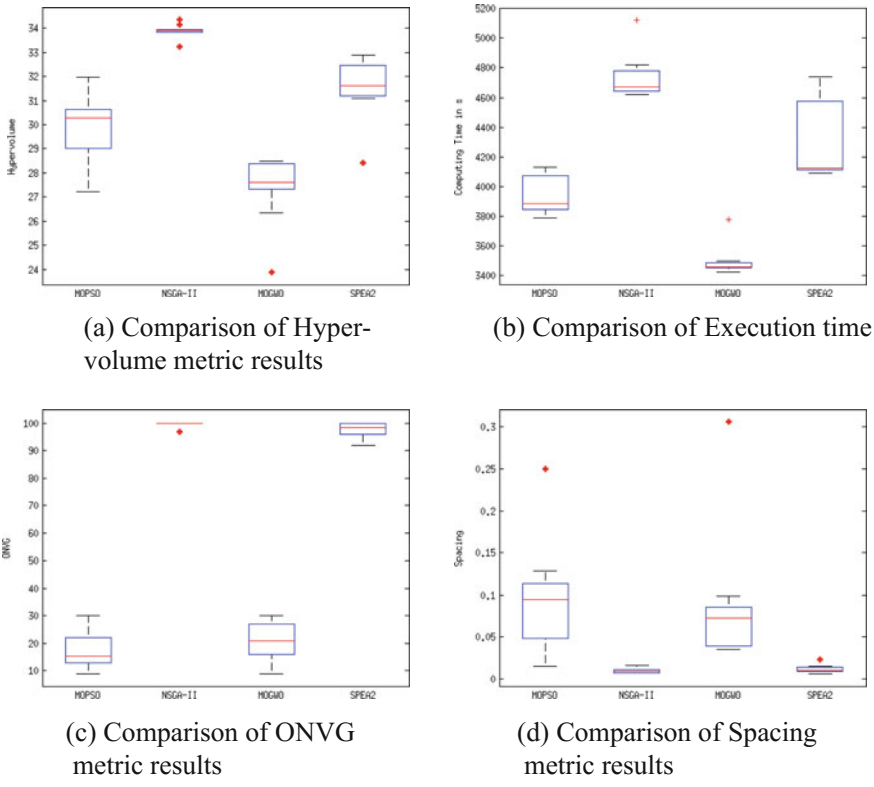


Fig. 1 Pareto front of *MOPSO*, *NSGA − II*, *MOGWO*, and *SPEA*2 obtained after 500 iterations for a population of 100 individuals



(a) Comparison of Hypervolume metric results

(b) Comparison of Execution time

(c) Comparison of ONVG metric results

(d) Comparison of Spacing metric results

Fig. 2 Comparison of the results obtained in 10 different runs by the four algorithms with a population of 100 individuals and after 500 generations

was produced by running each MOEA algorithm ten times with a population of 100 individuals and a maximum of 500 generations. From the figure we can see that the solution quality with respect to Pareto optimality obtained by MOGWO was quite low. MOPSO, NSGA-II, and SPEA2 produced a PF with similar patterns but different values. For this test, the NSGA-II algorithm had a better convergence. Starting with a comparison of the hypervolume metric, it becomes clear that the NSGA-II PF score comes first, followed by SPEA2, MOPSO, and finally MOGWO as shown in Fig. 2a. A similar order is also obtained with respect to ONVG and spacing metrics as indicated in Fig. 2c, d. As for computational time, the boxplot in Fig. 2b clearly indicates that MOGWO outperformed all algorithms while NSGA-II scored last.

5 Conclusion

The multioptimization method developed in this study can provide an efficient strategy for the radars optimal placement problem, resulting in network configurations at a relatively short time and with sufficient accuracy. For our study region, the proposed strategy gave results that were relatively insensitive to the number of individuals in the population of MOEA involved in the selection of a single best network. The radar coverage and cost objective functions selected for this study appear to be suitable for guiding network selection in support for a better weather observation. This tool could reduce valuable time and cost through the reduction of suitable sites that are evaluated on field by experts.

References

1. Coello Coello, C.A., Lechuga, M.: Mopso: a proposal for multiple objective particle swarm optimization. In: Proceedings of the 2002 Congress on Evolutionary Computation, 2002. CEC 2002, vol. 2, pp. 1051–1056. doi:10.1109/CEC.2002.1004388
2. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-ii. IEEE Trans. Evol. Comput. **6**(2), 182–197 (2002). doi:10.1109/4235.996017
3. Doviak, R., Zrnić, D.: Doppler Radar and Weather Observations. Academic Press, San Diego (1984)
4. Kurdzo, J.M., Palmer, R.D.: Objective optimization of weather radar networks for low-level coverage using a genetic algorithm. J. Atmos. Ocean. Technol. **29**(6), 807–821 (2012)
5. Minciardi, R., Sacile, R., Siccardi, F.: Optimal planning of weather radar network. J. Atmos. Ocean. Technol. **20**, 1251–1262 (2003)
6. Mirjalili, S., Saremi, S., Mirjalili, S.M., dos Coelho, S.L.: Multi-objective grey wolf optimizer: a novel algorithm for multi-criterion optimization. Expert Syst. Appl. **47**, 106–119 (2016)
7. Schott, J.R.: Fault Tolerant Design Using Single and Multicriteria Genetic Algorithm Optimization. Master's thesis, Massachusetts Institute of Technology (1995)
8. Veldhuizen, D.A.V., Lamont, G.B.: On measuring multiobjective evolutionary algorithm performance. In: Proceedings of the 2000 Congress on Evolutionary Computation, 2000, vol. 1, pp. 204–211. doi:10.1109/CEC.2000.870296
9. Whiton, R.C., Smith, P.L., Bigler, S.G., Wilk, K.E., Harbuck, A.C.: History of operational use of weather radar by U.S. weather services. Part ii: Development of operational doppler weather radars, p. 244 (1998)
10. Wilson, J., Carbone, R., Boynton, H., Serafin, R.: Operational application of meteorological doppler radar. Bull. Am. Meteorol. Soc. **61**, 1154–1168 (1980)
11. Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. IEEE Trans. Evol. Comput. **3**(4), 257–271 (1999). doi:10.1109/4235.797969
12. Zitzler, E., Giannakoglou, K., Tsahalis, D., Periaux, J., Papailiou, K.: SPEA2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: TF, Ler, E.Z., Laumanns, M., Thiele, L. (eds.) (2002)

Building Decision Making Models Through Conceptual Constraints: Multi-scale Process Model Implementations

Canan Dombayci and Antonio Espuña

Abstract The integration of decision-making procedures typically assigned to different hierarchical levels in a production system (strategic, tactical, and operational) requires the use of complex multi-scale mathematical models and high computational efforts, in addition to the need of an extensive management of data and knowledge within the production system. The aim of this study is to propose a comprehensive solution for this integration problem through the use of Conceptual Constraints. The presented methodology is based on a model in a domain ontology and the use of generalized concepts to develop tailor-made decision making models, created according to the introduced data. Different decision making formulations are reviewed and, accordingly, comprehensive Conceptual Constraints for the different concepts (like material balances) can be determined. This work shows how these Conceptual Constraints can be used when the quality of information is changed, enabling multi-scale implementations.

1 Introduction

The Committee on Challenges for the Chemical Sciences in the 21st Century [1] indicates that the development of new and powerful computational methods, applicable from the atomic level to the chemical process and enterprise levels, is a key factor to enable multi-scale optimization. This would broaden the scope of one of the main objectives attained by the Process Systems Engineering (PSE) approach, focused on the systematization of the decision making through modeling and optimization, to a new generalized paradigm. In this line, Harjunkoski et al. [4] address the usage of standards to systematically build models and to be able to create a master model to configure new problems without modifying the algorithmic core, or Hooker [5]

C. Dombayci (✉) · A. Espuña
Chemical Engineering Department, EEBE, Universitat Politècnica de Catalunya,
C. Eduard Maristany, 10-14, 08019 Barcelona, Spain
e-mail: canan.dombayci@upc.edu

A. Espuña
e-mail: antonio.espuna@upc.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_12

uses metaconstraints through the use of a pre-built library, in order to assist model builders in a constraint-programming framework. However, although the practical implementations based on these approaches introduce significant improvements during model building, these constraints are not connected conceptually to problems to be solved in the system and the complete model building for the integration problem is not investigated.

Therefore, this work investigates systematic model building procedures to address optimization problems from a multi-scale perspective and to automatically generate the problem instances according to the problem to be solved.

2 Analysis of Conceptual Constraints

The traditional modeling approach is based on the following steps: (i) analysis of the process, (ii) conceptual model of the process, (iii) mathematical representation of the problem, and (iv) iterative model improvements [8]. Usually, the model of the process is based on mathematical expressions related to fundamental laws such as balances, sequencing and allocation constraints. Then, other constraints according to the details of the problem are added; for instance: in short-term scheduling models, time constraints can be used to describe shifts or maintenance requirements [9]. Afterward, the constraints are detailed according to the model granularity (e.g.: the used time representation), the given data and other presented details of the requirements. Since these formulations are constructed specifically for a problem, they remain static with the given data structure and model construction, and can not be reused at different levels even within the same organization.

In order to overcome these limitations, it is proposed to aggregate the abstract information related to a common concept, to be used at different hierarchical levels to create a Conceptual Constraint (CC) Domain. Then, this CC Domain may be used to create upper level relations and may be connected with different sets of data available in the production system in the PSE Domain. Figure 1 shows the connections between two domains with the CurrentlyAvailableMaterial[1] example.

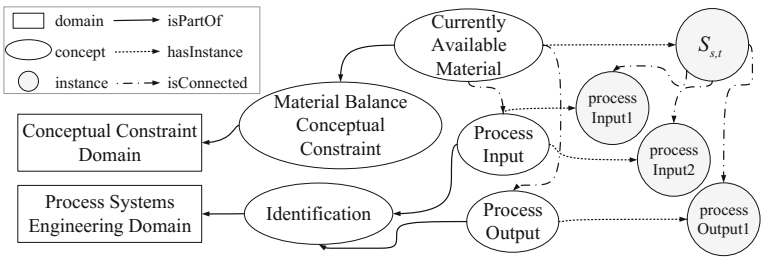


Fig. 1 Proposed modeling approach

[1]Concept names are written using CamelCase representation.

The CurrentlyAvailableMaterial concept is part of the MaterialBalance CC and it has different instance connections which link the CC Domain to the PSE Domain. An illustrative example of the MaterialBalance CC and these connections are given in Sect. 3 showing these connections.

Based on this idea, the proposed modeling approach exploits the CC to formulate the problem at a higher (more generic) level, which is dynamically connected to the data in the PSE Domain. CCs actually represent the main principles of the technological system (like, for example, the material balances—Fig. 2). Then, to create the problem instance to be solved, the elements used to represent this main principle (following the same example, the CurrentlyAvailableMaterial are connected to ProcessInput and ProcessOutput concepts, which are part of the Identification concept in the PSE Domain. These concepts are gathered as Identification concept since ProcessInput is defined as an identification of materials, energy, or other resources required for a recipe.

There are two main aspects to be emphasized in this new way to approach the model construction. The first one is related to the way how some knowledge is managed to identify where the inputs of the system are loaded into the ontological model [3]. The required systematic approach will typically imply the standardization of the information; in this work, the ISA proposals (ISA88 and ISA95 Standards) have been applied, so the models include the recipe, the procedural model, and the physical model. The resulting ontological model is represented by the PSE Domain in Fig. 1 (interested reader is referred to [2] for a detailed explanation). The second one is the constraint management associated to the connection of the two domains. The CC Domain elements construct the problem formulation considering the PSE Domain, and the suggested methodology simply implements the following steps: (i) ontological representation of the problem in the PSE Domain, (ii) selection from the CCs, (iii) model creation from the CCs and introduced data, and (iv) solution of the model.

Furthermore, the claim is that CCs are not only applicable to a certain hierarchical level (like strategic versus tactical level). The same concept appears at different levels with different information and assumptions. Therefore, this approach uses some generic concept connections in order to identify equivalences in different hierarchical levels. For instance, in the case of a material balance, depending on the available information, it can be constructed around a unit or a site and the process inputs and outputs will change, accordingly (Sect. 3).

3 Application: Material Balance Conceptual Constraint

Because of the space limitations, only the construction of one CC is detailed in this paper. The physical model is limited to units and sites. In order to explain CCs, three material balance equations are taken from the literature [6, 7]. The first constraint is given in Fig. 2, which belongs to a short-term scheduling formulation [6] and the

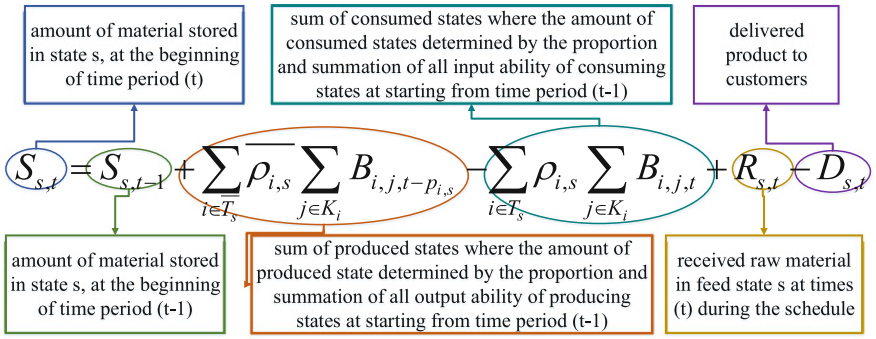
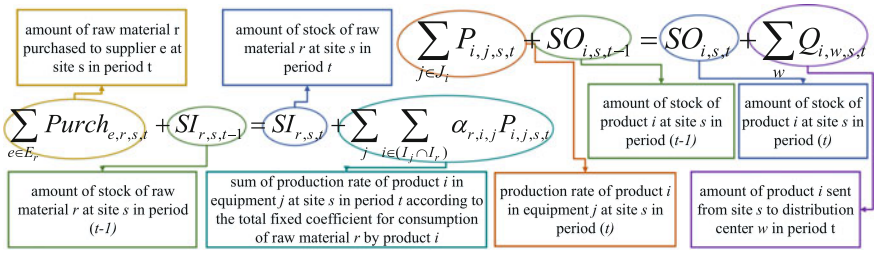


Fig. 2 Material balance from short-term scheduling formulations [6]



(a) For raw materials [7] (b) For products [7]

Fig. 3 Material balances from planning formulation

other two equations, depicted in Fig. 3a, b, represent the material balance developed and used in a planning formulation [7].

In the figures, each element of the constraints is examined semantically and described in the attached text-boxes according to the corresponding nomenclature.[2]

In the planning formulation [7], the material balance constraints for raw materials and products are created, separately. The first observation for the material balances in Fig. 3 is that this separation can be overcome using the recipe concept which is also known as state-task network representation [6]. When the planning [7] and the scheduling formulations [6] are compared, the variable related to the production uses different physical elements: sites and units, respectively. In order to integrate different levels, differentiation of the physical and procedural models are required, which is partially given in ISA88 Batch Control Standard and applicable to other operation modes.

Combining the three examined equations gives the general view of the elements in the material balance CC. This general view contains the intermediate part of the constraint construction. Figure 4 summarizes the final generic mathematical equation instances and their connection to the elements in the material balance CC. The

[2]Check the original sources for a detailed description of the nomenclature used in these equations.

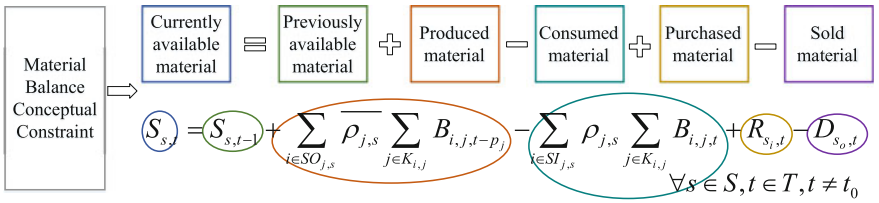


Fig. 4 Material Balance CC and resulted mathematical expression

Table 1 Nomenclature for the material balance CC:

Sets	Member concepts	Subsets	Explanation
s	Process, Site Process Segment Input, Process, Site Process Segment Output	$K_{i,j}$	Mapping between physical and procedural model
j	Unit Procedure, Site Procedure	*SO* & *SI*	Recipe connection
i	Unit, Site	s_i	Process, Site Process Segment Input
t	Time period	s_o	Process, Site Process Segment Output
Parameters	Explanation	Variables	Explanation
$\rho_{j,s}$	The proportion of input	$S_{s,t}$	Currently available material
$\overline{\rho}_{j,s}$	The proportion of output	$B_{i,j,t}$	Undertaken material for production
p_j	Processing time of the procedural model elements		

hierarchical conceptual relations of the constraint are given in Table 1. Relations in this paper are restricted to the Unit and Site levels in the hierarchy. While Fig. 2 has instances from the Unit to be used as set, equations in Fig. 3 have the Site concept instances. So, when a problem is required to be solved at Unit level, the same Conceptual Concept, depicted in Fig. 4, is applied at the Unit level. Also the Site concept instances are called when a problem at the Site level is required to be solved.

An additional example would be the CurrentlyAvailableMaterial concept, which is connected with an Identification concept to get the ProcessInput and the ProcessOutput for the identified level Fig. 1. In the case of the planning model, the Identification concept, which describes materials required for recipes, includes the SiteProcessSegmentInput (raw materials) and SiteProcessSegmentOutput (products) concepts. Then, the CurrentlyAvailableMaterial may become a function of

$$CurrentlyAvailableMaterial_{(Identification,PhysicalModel,Time)}$$

where the Identification refers to set of materials depending on the level. The PhysicalModel element includes the set of Unit or Site and the Time element adds the information related to the discretization (if the formulation is a discrete time).

4 Conclusions

This paper presents a methodology for building mathematical models from existing data using Conceptual Constraints (CCs). The aim of the study is to be able to comprehensively formulate and solve decision making problems from different points of view in a production system using a multi-scale generic approach. As a motivating example, the material balance has been selected to illustrate the use of a common CC at different decision-making levels. When some specific data-set related to the problem is selected, it is connected with the CC and the model structure is automatically generated from them. The proposed methodology is applicable to any system where a set of rules regulating the relations (connections) between the different sub-systems exists, provided that the information inside these systems is modeled accordingly. In the case of multi-level hierarchical systems, these relations are clear, previously identified and even standardized, so the application of the proposed methodology and the identification of the conceptual equivalences becomes evident; in the case of other systems, such as interwoven systems, systems of systems, etc., the relations may be more difficult to standardize for a generic case, although common concepts will also exist and might be exploited accordingly. As a result, and obviously accepting that there will be always constraints which are not practical or feasible to generalize, this methodology provides a basis for the systematic creation of models and, even more important, to ensure the coherence of the results obtained by different models operating at different hierarchical levels in a multi-scale system.

Acknowledgements Financial support from the Spanish Ministry of Economy and Competitiveness, ECOCIS (DPI2013-48243-C2-1-R), AGAUR and the European Regional Development Fund, 2014-SGR-1092-CEPEiMA is fully appreciated.

References

1. Breslow, R., Tirrell, M.V.: Beyond the Molecular Frontier, Challenges for Chemistry and Chemical Engineering, Committee on Challenges for the Chemical Sciences in the 21st Century. The National Academies Press (2003)
2. Dombayci, C., Farreres, J., Rodríguez, H., Muñoz, E., Capón-García, E., Espuña, A., Graells, M.: On the process of building a process systems engineering ontology using a semi-automatic construction approach. Comput. Aided Chem. Eng. **37**, 941–946 (2015)
3. Dombayci, C., Medina, S., Graells, M., Espuña, A.: Integrated management of hierarchical levels: towards a CAPE tool. In: Kravanja, Z. (ed.) Comput. Aided Chem. Eng., pp. 7–12 (2016)

4. Harjunkoski, I., Maravelias, C.T., Bongers, P., Castro, P.M., Engell, S., Grossmann, I.E., Hooker, J., Méndez, C., Sand, G., Wassick, J.: Scope for industrial applications of production scheduling models and solution methods. Comput. Chem. Eng. **62**, 161–193 (2014)
5. Hooker, J.N.: Integrated Methods for Optimization, International Series in Operations Research & Management Science, vol. 170. Springer, Boston (2012)
6. Kondili, E., Pantelides, C., Sargent, R.: A general algorithm for short-term scheduling of batch operations—I. MILP formulation. Comput. Chem. Eng. **17**(2), 211–227 (1993)
7. Laínez, J.M., Guillén-Gosálbez, G., Badell, M., Espuña, A., Puigjaner, L.: Enhancing corporate value in the optimal design of chemical supply chains. Ind. Eng. Chem. Res. **46**(23), 7739–7757 (2007)
8. Makowski, M.: A structured modeling technology. Eur. J. Oper. Res. **166**(3), 615–648 (2005)
9. Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M.: State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. **30**(6–7), 913–946 (2006)

Methods of Tropical Optimization in Rating Alternatives Based on Pairwise Comparisons

Nikolai Krivulin

Abstract We apply methods of tropical optimization to handle problems of rating alternatives on the basis of the log-Chebyshev approximation of pairwise comparison matrices. We derive a direct solution in a closed form, and investigate the obtained solution when it is not unique. Provided the approximation problem yields a set of score vectors, rather than a unique (up to a constant factor) one, we find those vectors in the set, which least and most differentiate between the alternatives with the highest and lowest scores, and thus can be representative of the entire solution.

1 Introduction

Tropical (idempotent) mathematics, which deals with the theory and applications of semirings with idempotent addition [4, 6], finds use in operations research, computer science and other fields. Optimization problems that are formulated and solved in the framework of tropical mathematics constitute an important research domain, which offers new solutions to old and novel problems in various applied areas, including project scheduling [7, 10], location analysis [9] and decision making [8, 11]. The problems are usually defined to minimize or maximize functions on vectors over idempotent semifields (semirings with multiplicative inverses).

In this paper, we apply methods of tropical optimization to handle problems of rating alternatives on the basis of the log-Chebyshev approximation of pairwise comparison matrices. We derive a direct solution in a closed form, and investigate the solution when it is not unique. Provided the approximation problem yields a set of score vectors, rather than a unique (up to a constant factor) one, we find those vectors in the set, which least and most differentiate between the alternatives with the highest and lowest scores, and thus can be representative of the entire solution.

N. Krivulin (✉)
St. Petersburg State University, 7/9 Universitetskaya nab.,
St. Petersburg 199034, Russia
e-mail: nkk@math.spbu.ru

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_13

2 Rating Alternatives via Pairwise Comparisons

The method of rating alternatives from pairwise comparisons finds use in decision making when a direct evaluation of the ratings is unacceptable or infeasible (see, e.g., [12] for further details). The outcome of the comparisons is described by a square symmetrically reciprocal matrix $\mathbf{A} = (a_{ij})$, where a_{ij} shows the relative preference of alternative i over j, and satisfies the condition $a_{ij} = 1/a_{ji} > 0$ for all i, j.

To provide consistency of the data given by pairwise comparison matrices, the entries of the matrices must be transitive to provide the equality $a_{ij} = a_{ik}a_{kj}$ for all i, j, k. A pairwise comparison matrix with only transitive entries is called consistent.

For each consistent matrix $\mathbf{A} = (a_{ij})$, there is a positive vector $\mathbf{x} = (x_i)$ whose elements completely determine the entries of $\mathbf{A}$ by the relation $a_{ij} = x_i/x_j$. Provided that a matrix $\mathbf{A}$ is consistent, its corresponding vector $\mathbf{x}$ is considered to represent directly, up to a positive factor, the individual scores of alternatives in question.

The pairwise comparison matrices encountered in practice are generally inconsistent, which leads to a problem of approximating these matrices by consistent matrices. To solve the problem, the approximation with the principal eigenvector [12, 13], least squares approximation [2, 13] and other techniques [1, 3, 5] are used.

Another approach involves the approximation of a reciprocal matrix $\mathbf{A} = (a_{ij})$ by a consistent matrix $\mathbf{X} = (x_{ij})$ in the log-Chebyshev sense, where the approximation error is measured with the Chebyshev metric on the logarithmic scale. Since both matrices $\mathbf{A}$ and $\mathbf{X}$ have positive entries, and the logarithm is monotone increasing, the error can be written as $\max_{i,j} |\log a_{ij} - \log x_{ij}| = \log \max_{i,j} \max\{a_{ij}/x_{ij}, x_{ij}/a_{ij}\}$.

Considering that the minimization of the logarithm is equivalent to minimizing its argument, and that the matrix $\mathbf{X}$ can be defined through a positive vector $\mathbf{x} = (x_i)$ by the equality $x_{ij} = x_i/x_j$ for all i, j, the error function to minimize is replaced by $\max_{i,j} \max\{a_{ij}/x_{ij}, x_{ij}/a_{ij}\} = \max_{i,j} \max\{a_{ij}x_j/x_i, a_{ji}x_i/x_j\}$. The application of the condition $a_{ij} = 1/a_{ji}$ yields $\max_{i,j} \max\{a_{ij}x_j/x_i, a_{ji}x_i/x_j\} = \max_{i,j} a_{ij}x_j/x_i$, which finally reduces the approximation problem to finding positive vectors $\mathbf{x}$ to

$$\text{minimize} \quad \max_{i,j} a_{ij}x_j/x_i. \tag{1}$$

Assume that the approximation results in a set $\mathscr{S}$ of score vectors $\mathbf{x}$, rather than a unique (up to a constant factor) one. Then, further analysis is needed to reduce to a very few representative solutions, such as some "worst" and "best" solutions.

As the purpose of calculating the scores is to differentiate alternatives, one can concentrate on two vectors $\mathbf{x} = (x_i)$ from $\mathscr{S}$, which least and most differentiate between the alternatives with the highest and lowest scores by minimizing and maximizing the contrast ratio $\max_i x_i / \min_i x_i = \max_i x_i \cdot \max_i x_i^{-1}$. Then, the problem of calculating the least (the most) differentiating solution is to find vectors $\mathbf{x} \in \mathscr{S}$ that

$$\text{minimize (maximize)} \quad \max_i x_i \cdot \max_i x_i^{-1}. \tag{2}$$

Below, we reformulate problems (1) and (2) in terms of tropical mathematics, and then apply recent results in tropical optimization to offer complete, direct solutions.

3 Preliminary Definitions, Notation and Results

We start with a brief overview of the basic definitions and notation of tropical algebra. For further details on tropical mathematics, see, e.g., recent publications [4, 6].

Consider the set of nonnegative reals $\mathbb{R}_+$, which is equipped with two operations, addition $\oplus$ defined as maximum, and multiplication $\otimes$ defined as usual, and has 0 and 1 as their neutral elements. Addition is idempotent, since $x \oplus x = \max(x, x) = x$ for all $x \in \mathbb{R}_+$. Multiplication is distributive over addition and invertible to give each $x \neq 0$ an inverse x^{-1} such that $x \otimes x^{-1} = xx^{-1} = 1$. The system $(\mathbb{R}_+, \oplus, \otimes, 0, 1)$ is called the idempotent semifield or the max-algebra and denoted $\mathbb{R}_{\max}$. In the sequel, the sign $\otimes$ is omitted for brevity. The power notation has the standard meaning.

The set of matrices over $\mathbb{R}_+$ with m rows and n columns is denoted by $\mathbb{R}_+^{m\times n}$. A matrix with all zero entries is the zero matrix. The matrices without zero rows are called row-regular. Matrix operations employ the conventional entry-wise formulae, where the scalar operations $\oplus$ and $\otimes$ play the role of the usual addition and multiplication.

The multiplicative conjugate transpose of a nonzero matrix $\mathbf{A} = (a_{ij})$ is the matrix $\mathbf{A}^- = (a_{ij}^-)$ with the entries $a_{ij}^- = a_{ji}^{-1}$ if $a_{ji} \neq 0$, and $a_{ij}^- = 0$ otherwise.

Consider the square matrices in the set $\mathbb{R}_+^{n\times n}$. A matrix with 1 along the diagonal and 0 elsewhere is the identity matrix denoted $\mathbf{I}$. The power notation specifies iterated products as $\mathbf{A}^0 = \mathbf{I}$ and $\mathbf{A}^p = \mathbf{A}^{p-1}\mathbf{A}$ for any matrix $\mathbf{A}$ and integer $p > 0$.

The tropical spectral radius of a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}_+^{n\times n}$ is the scalar given by

$$\lambda = \bigoplus_{1\leq k\leq n} \bigoplus_{1\leq i_1,\ldots,i_k\leq n} (a_{i_1i_2}a_{i_2i_3}\cdots a_{i_ki_1})^{1/k}. \tag{3}$$

The asterate operator (the Kleene star) maps the matrix $\mathbf{A}$ onto the matrix

$$\mathbf{A}^* = \mathbf{I} \oplus \mathbf{A} \oplus \cdots \oplus \mathbf{A}^{n-1}. \tag{4}$$

The column vectors with n elements form the set $\mathbb{R}_+^n$. The vectors with all elements equal to 0 and to 1 are denoted by $\mathbf{0}$ and $\mathbf{1}$. A vector is regular if it has no zero elements. For any nonzero column vector $\mathbf{x} = (x_i)$, its conjugate transpose is the row vector $\mathbf{x}^- = (x_i^-)$, where $x_i^- = x_i^{-1}$ if $x_i \neq 0$, and $x_i^- = 0$ otherwise.

We conclude the overview with examples of tropical optimization problems. Suppose that, given a matrix $\mathbf{A} = (a_{ij}) \in \mathbb{R}_+^{n\times n}$, we need to find vectors $\mathbf{x} \in \mathbb{R}_+^n$ that

$$\text{minimize} \quad \mathbf{x}^-\mathbf{A}\mathbf{x}. \tag{5}$$

The next complete, direct solution to the problem is obtained in [7].

Lemma 1 *Let* $\mathbf{A}$ *be a matrix with spectral radius* $\lambda > 0$*. Then, the minimum value in* (5) *is equal to* λ*, and all regular solutions are given by* $\mathbf{x} = (\lambda^{-1}\mathbf{A})^{*}\mathbf{u}$, $\mathbf{u} \neq \mathbf{0}$.

Given a matrix $\mathbf{A} \in \mathbb{R}_{+}^{m \times n}$ and vectors $\mathbf{p} \in \mathbb{R}_{+}^{m}$, $\mathbf{q} \in \mathbb{R}_{+}^{n}$, we now find $\mathbf{x} \in \mathbb{R}_{+}^{n}$ that

$$\text{minimize} \quad \mathbf{q}^{-}\mathbf{x}(\mathbf{A}\mathbf{x})^{-}\mathbf{p}. \tag{6}$$

A solution given by [9] uses a sparsification technique to provide the next result.

Lemma 2 *Let* $\mathbf{A} = (a_{ij})$ *be a row-regular matrix,* $\mathbf{p} = (p_i)$ *be nonzero and* $\mathbf{q} = (q_j)$ *be regular vectors, and* $\Delta = (\mathbf{A}\mathbf{q})^{-}\mathbf{p}$*. Let* $\widehat{\mathbf{A}} = (\widehat{a}_{ij})$ *denote the matrix with entries* $\widehat{a}_{ij} = a_{ij}$ *if* $a_{ij} \geq \Delta^{-1} p_i q_j^{-1}$*, and* $\widehat{a}_{ij} = 0$ *otherwise. Let* $\mathscr{A}$ *be the set of matrices obtained from* $\widehat{\mathbf{A}}$ *by fixing one nonzero entry in each row and setting the others to* 0*.*

Then, the minimum value in problem (6) *is equal to* $\Delta = (\mathbf{A}\mathbf{q})^{-}\mathbf{p}$*, and all regular solutions are given by the conditions* $\mathbf{x} = (\mathbf{I} \oplus {}^{-1}\mathbf{A}_{\mathbf{1}}^{-}\mathbf{p}\mathbf{q}^{-})\mathbf{u}$, $\mathbf{u} \neq \mathbf{0}$, $\mathbf{A}_{\mathbf{1}} \in \mathscr{A}$.

Finally, we consider a maximization version of problem (6) to find vectors $\mathbf{x}$ that

$$\text{maximize} \quad \mathbf{q}^{-}\mathbf{x}(\mathbf{A}\mathbf{x})^{-}\mathbf{p}. \tag{7}$$

A complete solution to the problem is obtained in [10]. Below, we describe this solution in a more compact vector form using the representation lemma in [9].

Lemma 3 *Let* $\mathbf{A} = (\mathbf{a}_j)$ *be a matrix with regular columns* $\mathbf{a}_j = (a_{ij})$*, and* $\mathbf{p} = (p_i)$ *and* $\mathbf{q} = (q_j)$ *be regular vectors. Let* $\mathbf{A}_{sk}$ *denote the matrix obtained from* $\mathbf{A}$ *by fixing the entry* a_{sk} *for some indices* s *and* k*, and replacing the other entries by* 0*.*

Then, the maximum value in (7) *is equal to* $\Delta = \mathbf{q}^{-}\mathbf{A}^{-}\mathbf{p}$*, and all regular solutions are given by* $\mathbf{x} = (\mathbf{I} \oplus \mathbf{A}_{sk}^{-}\mathbf{A})\mathbf{u}$, $\mathbf{u} \neq \mathbf{0}$, $k = \arg\max_j q_j^{-1}\mathbf{a}_j^{-}\mathbf{p}$, $s = \arg\max_i a_{ik}^{-1} p_i$.

4 Application to Rating Alternatives

We are now in a position to represent optimization problems (1) and (2) stated above in the tropical mathematics setting, and then to solve them in an explicit form.

Consider problem (1) of evaluating the score vector based on the log-Chebyshev approximation of a pairwise comparison matrix $\mathbf{A}$. In terms of the max-algebra $\mathbb{R}_{\max}$ the problem takes the form (5). Application of Lemma 1 yields the following result.

Theorem 1 *Let* $\mathbf{A}$ *be a pairwise comparison matrix with spectral radius* λ*, and denote* $\mathbf{A}_{\lambda} = \lambda^{-1}\mathbf{A}$ *and* $\mathbf{B} = \mathbf{A}_{\lambda}^{*}$*. Then, all score vectors are given by* $\mathbf{x} = \mathbf{B}\mathbf{u}$, $\mathbf{u} \neq \mathbf{0}$.

Example 1 Suppose the result of comparing $n = 4$ alternatives is given by the matrix

$$\mathbf{A} = \begin{pmatrix} 1 & 1/3 & 1/2 & 1/3 \\ 3 & 1 & 4 & 1 \\ 2 & 1/4 & 1 & 2 \\ 3 & 1 & 1/2 & 1 \end{pmatrix}. \tag{8}$$

To apply Theorem 1, we use (3) to find $\lambda = (a_{23}a_{34}a_{42})^{1/3} = 2$, and calculate $\mathbf{A}_\lambda = \begin{pmatrix} 1/2 & 1/6 & 1/4 & 1/6 \\ 3/2 & 1/2 & 2 & 1/2 \\ 1 & 1/8 & 1/2 & 1 \\ 3/2 & 1/2 & 1/4 & 1/2 \end{pmatrix}$. Then, we follow (4) to compute $\mathbf{A}_\lambda^* = \begin{pmatrix} 1 & 1/6 & 1/3 & 1/3 \\ 3 & 1 & 2 & 2 \\ 3/2 & 1/2 & 1 & 1 \\ 3/2 & 1/2 & 1 & 1 \end{pmatrix}$.

As the last three columns of the matrix $\mathbf{A}_\lambda^*$ are collinear, we take one of them, say, the second. Combining with the first column multiplied by $1/3$ leads to the solution

$$\mathbf{x} = \mathbf{B}\mathbf{u}, \qquad \mathbf{B} = \begin{pmatrix} 1/3 & 1/6 \\ 1 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}, \qquad \mathbf{u} = (u_1, u_2)^T, \qquad u_1, u_2 \neq 0. \tag{9}$$

Note that all the solutions assign the highest score to the second alternative and the lowest to the first. Moreover, the solutions which least and most differentiate between these alternatives, are the first and the second columns in the matrix $\mathbf{B}$.

In the general case, the least and most differentiating solutions from a set of vectors, given in the form $\mathbf{x} = \mathbf{B}\mathbf{u}$, are determined by solving problems (2). The problems are to minimize and maximize the contrast ratio for the elements of the vector $\mathbf{x}$, which, in terms of tropical mathematics, takes the form $\mathbf{1}^T\mathbf{x}\mathbf{x}^{-}\mathbf{1} = \mathbf{1}^T\mathbf{B}\mathbf{u}(\mathbf{B}\mathbf{u})^{-}\mathbf{1}$.

To find a vector $\mathbf{x} = \mathbf{B}\mathbf{u}$ with the least differentiation between scores, we solve the problem

$$\text{minimize} \quad \mathbf{1}^T\mathbf{B}\mathbf{u}(\mathbf{B}\mathbf{u})^{-}\mathbf{1}.$$

Assuming the matrix $\mathbf{B}$ is obtained as in Theorem 1, we have the next result.

Theorem 2 *Let $\widehat{\mathbf{B}}$ be a sparsified matrix derived from $\mathbf{B}$ by setting to 0 all entries below $\Delta^{-1} = ((\mathbf{B}(\mathbf{1}^T\mathbf{B})^{-})^{-}\mathbf{1})^{-1}$, and $\mathscr{B}$ be the set of matrices obtained from $\widehat{\mathbf{B}}$ by fixing one nonzero entry in each row and setting the others to 0. Then, the least differentiating score vectors are given by $\mathbf{x} = \mathbf{B}(\mathbf{I} \oplus \Delta^{-1}\mathbf{B}_1^{-}\mathbf{1}\mathbf{1}^T\mathbf{B})\mathbf{v}$, $\mathbf{v} \neq \mathbf{0}$, $\mathbf{B}_1 \in \mathscr{B}$.*

Proof We reduce the problem under study to (6) by the substitutions $\mathbf{q}^{-} = \mathbf{1}^T\mathbf{B}$, $\mathbf{A} = \mathbf{B}$, $\mathbf{p} = \mathbf{1}$ and $\mathbf{x} = \mathbf{u}$. Since the matrix $\mathbf{B}$ has only nonzero entries, the regularity conditions of Lemma 2 are satisfied. Application of this lemma involves evaluating the minimum value $\Delta = (\mathbf{B}(\mathbf{1}^T\mathbf{B})^{-})^{-}\mathbf{1}$, calculating the sparsified matrix $\widehat{bfB}$, and forming the matrix set $\mathscr{B}$. The solution is given by $\mathbf{u} = (\mathbf{I} \oplus \Delta^{-1}\mathbf{B}_1^{-}\mathbf{1}\mathbf{1}^T\mathbf{B})\mathbf{v}$, where $\mathbf{v} \neq \mathbf{0}$ and $\mathbf{B}_1 \in \mathscr{B}$. Turning back to the vector $\mathbf{x} = \mathbf{B}\mathbf{u}$ yields the desired result. □

Example 2 Consider the solution obtained in the form (9) in Example 1 for the matrix (8). To apply the result of Theorem 2, we successively calculate $\mathbf{1}^T\mathbf{B} = \begin{pmatrix} 1 & 1 \end{pmatrix}$, $\mathbf{B}(\mathbf{1}^T\mathbf{B})^{-} = \begin{pmatrix} 1/3 \\ 1 \\ 1/2 \\ 1/2 \end{pmatrix}$, $\Delta = (\mathbf{B}(\mathbf{1}^T\mathbf{B})^{-})^{-}\mathbf{1} = 3$, and $\widehat{\mathbf{B}} = \begin{pmatrix} 1/3 & 0 \\ 1 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$.

We now examine the matrices obtained from $\widehat{\mathbf{B}}$ by leaving one nonzero entry in each row. For instance, consider the matrix $\mathbf{B}_1 = \begin{pmatrix} 1/3 & 0 \\ 1 & 0 \\ 1/2 & 0 \\ 1/2 & 0 \end{pmatrix}$, which leaves the first column in $\widehat{\mathbf{B}}$ unchanged, and has all zero entries in the second. We have $\mathbf{B}_1^- \mathbf{1} = \begin{pmatrix} 3 \\ 0 \end{pmatrix}$, $\mathbf{B}_1^- \mathbf{1}\mathbf{1}^T \mathbf{B} = \begin{pmatrix} 3 & 3 \\ 0 & 0 \end{pmatrix}$, $\mathbf{I} \oplus \Delta^{-1} \mathbf{B}_1^- \mathbf{1}\mathbf{1}^T \mathbf{B} = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$, and $\mathbf{B}(\mathbf{I} \oplus \Delta^{-1} \mathbf{B}_1^- \mathbf{1}\mathbf{1}^T \mathbf{B}) = \begin{pmatrix} 1/3 & 1/3 \\ 1 & 1 \\ 1/2 & 1/2 \\ 1/2 & 1/2 \end{pmatrix}$.

As both columns in the last matrix coincide, we take one to write the least differentiating solution in the form $\mathbf{x} = \begin{pmatrix} 1/3 & 1 & 1/2 & 1/2 \end{pmatrix}^T v$, $v \neq 0$. Calculations with the other matrices obtained from $\widehat{\mathbf{B}}$ yield the same result, and are thus omitted.

To obtain the most differentiating score vectors we need to solve the problem

$$\text{maximize} \quad \mathbf{1}^T \mathbf{B}\mathbf{u}(\mathbf{B}\mathbf{u})^- \mathbf{1}.$$

Similarly as before, we reduce this problem to (7), conclude that the conditions of Lemma 3 are fulfilled, and finally apply this lemma to obtain the next solution.

Theorem 3 *Let* $\mathbf{B} = (\mathbf{b}_j)$ *be a matrix with columns* $\mathbf{b}_j = (b_{ij})$, *and* $\mathbf{B}_{sk}$ *denote the matrix obtained from* $\mathbf{B}$ *by fixing the entry* b_{sk} *and replacing the others by* 0.

Then, the most differentiating score vectors are given by $\mathbf{x} = \mathbf{B}(\mathbf{I} \oplus \mathbf{B}_{sk}^- \mathbf{B})\mathbf{v}$, $\mathbf{v} \neq \mathbf{0}$, $k = \arg\max_j \mathbf{1}^T \mathbf{b}_j \mathbf{b}_j^- \mathbf{1}$, $s = \arg\max_i b_{ik}^{-1}$.

Example 3 We start with the solution at (9), and compute $\mathbf{1}^T \mathbf{b}_1 = 1$, $\mathbf{1}^T \mathbf{b}_2 = 1$, $\mathbf{b}_1^- \mathbf{1} = 3$, and $\mathbf{b}_2^- \mathbf{1} = 6$. Since $\mathbf{1}^T \mathbf{b}_1 \mathbf{b}_1^- \mathbf{1} = 3$ and $\mathbf{1}^T \mathbf{b}_2 \mathbf{b}_2^- \mathbf{1} = 6$, we take $k = 2, s = 1$.

We have $\mathbf{B}_{12} = \begin{pmatrix} 0 & 1/6 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$, $\mathbf{I} \oplus \mathbf{B}_{12}^- \mathbf{B} = \begin{pmatrix} 1 & 0 \\ 2 & 1 \end{pmatrix}$, and $\mathbf{B}(\mathbf{I} \oplus \mathbf{B}_{12}^- \mathbf{B}) = \begin{pmatrix} 1/3 & 1/6 \\ 2 & 1 \\ 1 & 1/2 \\ 1 & 1/2 \end{pmatrix}$.

Since the columns in the last matrix are collinear, we take one of them, say, the second, to write the most differentiating vector as $\mathbf{x} = \begin{pmatrix} 1/6 & 1 & 1/2 & 1/2 \end{pmatrix}^T v$, $v \neq 0$.

Acknowledgements This work was supported in part by the Russian Foundation for Humanities (grant No. 16-02-00059). The author is very grateful to the referees for their valuable comments and suggestions, which have been incorporated into the revised version of the manuscript.

References

1. Barzilai, J.: Deriving weights from pairwise comparison matrices. J. Oper. Res. Soc. **48**(12), 1226–1232 (1997)
2. Chu, M.T.: On the optimal consistent approximation to pairwise comparison matrices. Linear Algebra Appl. **272**(1–3), 155–168 (1998)
3. Farkas, A., Lancaster, P., Rózsa, P.: Consistency adjustments for pairwise comparison matrices. Numer. Linear Algebra Appl. **10**(8), 689–700 (2003)
4. Golan, J.S.: Semirings and Affine Equations Over Them, Mathematics and Its Applications, vol. 556. Springer, New York (2003)

5. González-Pachón, J., Rodríguez-Galiano, M.I., Romero, C.: Transitive approximation to pairwise comparison matrices by using interval goal programming. J. Oper. Res. Soc. **54**(5), 532–538 (2003)
6. Heidergott, B., Olsder, G.J., van der Woude, J.: Max Plus at Work. Princeton Series in Applied Mathematics. Princeton University Press, Princeton (2006)
7. Krivulin, N.: Extremal properties of tropical eigenvalues and solutions to tropical optimization problems. Linear Algebra Appl. **468**, 211–232 (2015)
8. Krivulin, N.: Rating alternatives from pairwise comparisons by solving tropical optimization problems. In: Tang, Z., Du, J., Yin, S., He, L., Li, R. (eds.) 2015 12th International Conference on Fuzzy Systems and Knowledge Discovery (FSKD), pp. 162–167. IEEE (2015)
9. Krivulin, N.: Solving a tropical optimization problem via matrix sparsification. In: Kahl, W., Winter, M., Oliveira, J.N. (eds) Relational and Algebraic Methods in Computer Science, Lecture Notes in Computer Science, vol. 9348, pp. 326–343. Springer, Cham (2015)
10. Krivulin, N.: A maximization problem in tropical mathematics: a complete solution and application examples. Informatica **27**(3), 587–606 (2016)
11. Krivulin, N.: Using tropical optimization techniques to evaluate alternatives via pairwise comparisons. In: Gebremedhin, A.H., Boman, E.G., Ucar, B. (eds.) 2016 Proceedings of the 7th SIAM Workshop on Combinatorial Scientific Computing, pp. 62–72. SIAM, Philadelphia (2016)
12. Saaty, T.L.: On the measurement of intangibles: a principal eigenvector approach to relative measurement derived from paired comparisons. Not. Am. Math. Soc. **60**(2), 192–208 (2013)
13. Saaty, T.L., Vargas, L.G.: Comparison of eigenvalue, logarithmic least squares and least squares methods in estimating ratios. Math. Model. **5**(5), 309–324 (1984)

Part V
Discrete and Integer Optimization

New Constraints and Features for the University Course Timetabling Problem

M. Aschinger, S. Applebee, A. Bucur, H. Edmonds, P. Hungerländer and K. Maier

Abstract The university course timetabling problem deals with the task of scheduling lectures of a set of university courses into a given number of rooms and time periods, taking into account various hard and soft constraints. The goal of the International Timetabling Competitions ITC2002 and ITC2007 was to establish models for comparison that cover the most frequently found use cases. Our model, motivated by a project with University College London (UCL), builds on the standard model from track 3 of ITC2007. Compared to the standard model from the literature, we cover several new constraints and extra features. For example, we expand the ITC2007 framework to generate a timetable for several weeks of the term instead of only one and introduce the corresponding timetable regularity metric, which measures the consistency of time and room assignments for a course throughout the term. We suggest an Integer Linear Programming approach for solving this expanded timetabling problem and introduce a corresponding new benchmark library. Finally we conduct computational experiments and discuss the results obtained with respect to solution quality and practical suitability for UCL.

1 Introduction

The 2nd annual Timetabling Competition (ITC-2007) established a standardized framework for the timetabling problem in terms of problem formulation and test instances [5]. With knowledge of this, University College London (UCL) and Satalia (NPComplete Ltd.) funded the research presented in this paper to investigate the

This research was supported by the third party project *Satalia*, funded by Satalia (NPComplete Ltd.) and conducted at the Alpen-Adria-Universität Klagenfurt.

M. Aschinger · A. Bucur (✉) · P. Hungerländer · K. Maier
Department of Mathematics, Alpen-Adria Universität Klagenfurt, Klagenfurt, Austria
e-mail: pabucur@edu.aau.at

S. Applebee · H. Edmonds
Satalia (NPComplete Ltd.),
London, UK

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_14

value of optimization-based timetabling and to provide implementable results. The UCL timetabling problem was modelled as closely to true complexity as possible, using actual data from 2015 and new features and constraints not previously considered in the literature.

Reviewing the relevant literature we notice that while over the years local search techniques have dominated the field of timetabling research [3], recently new approaches based on SAT [1], Constraint Programming [6], Mathematical Programming [2] and Metaheuristics [4] have successfully entered the field. Our approach is based on the Mathematical Programming methodology, as we suggest an Integer Linear Programming (ILP) algorithm to solve the optimization problem at hand.

This paper is structured as follows. In Sect. 2, we describe the curriculum-based timetabling problem as it was proposed in the ITC-2007 competition and extend it to include some of the extra constraints and features required by UCL. In Sect. 3, we introduce our ILP approach and the way it encodes the different constraints. In Sect. 4 we introduce several relevant metrics and we present and discuss the results of our computational experiments. Section 5 concludes the paper.

2 Problem Formulation: UCL Extended Framework

The challenge of the curriculum-based course timetabling problem (CB-CTT) is to schedule lectures belonging to a set of courses $C = \{c_1, c_2, \dots, c_v\}$ to k periods $P = \{p_1, p_2, \dots, p_k\}$ and m rooms $R = \{r_1, r_2, \dots, r_m\}$, accounting at the same time for certain hard and soft constraints. In the CB-CTT the timetable is generated based upon a set of s university curricula $I = \{i_1, i_2, \dots, i_s\}$, to which the courses belong.

Next let us introduce assignment vectors q whose entries are set to 1, if a course-period-room combination is contained in timetable. Otherwise the entries are set to 0. Now a feasible solution of the problem is a binary vector q that satisfies all hard constraints. Finally the task of the CB-CTT is to find a vector q^* such that $f(q^*) \leqslant f(q)$ for all $q \in \tilde{q}$, where $f(\cdot)$ is an evaluation function summing up all violations of the soft constraints and $\tilde{q}$ denotes the set of all feasible assignment vectors. The exact problem formulation of the ITC-2007 framework and further details can be found in [5].

The UCL timetabling problem presents a wide range of challenges, since its features and additional constraints substantially exceed the ones from the ITC-2007 framework. Many of those features and additional constraints are omitted in this short paper due to space limitations and will be provided in a forthcoming paper. We include though the extensions that are most interesting from an academic viewpoint:

1. Our courses consist of activities with different durations, which relaxes the indistinguishability assumption of lectures from the literature. Therefore we define the set of all activities $A = \{a_1, a_2, \dots, a_n\}$ with corresponding durations of activities $d_a,\ a \in A$.
2. The UCL framework aims to generate feasible timetables for a set W of 10 consecutive weeks in a manner that guarantees the highest possible timetable regularity.

This means that whenever possible, activities should be scheduled in the same period and room over the different weeks.
3. Some activities within the same curriculum can be scheduled in parallel, e.g. if students need to attend only one of the practical lessons offered.
4. The activities have a specific predefined type, which must match the room type of the assigned room.

3 Integer Linear Programming Models

The Integer Linear Programming (ILP) solver presented in this paper is based on the approach suggested by [2]. The problem is split into two stages. In the first stage, each activity is assigned to an appropriate set of consecutive time periods. The assignment of activities to rooms is done in the second stage. Due to space limitation, we state the mathematical formulation only for the second stage and for the first stage we solely mention the most important newly developed constraints.

While our solver is optimized for the UCL framework, we have also tested it on the original ITC-2007 benchmark set. Preliminary results are very encouraging and hence we plan to provide a detailed analysis showing the competitiveness of our solver on the standard benchmarking sets from the literature in a forthcoming paper.

First Stage: In the first stage each activity has to be scheduled in a consecutive set of time periods. The function $D(p)$ gives the day of period p. Now if activity a is scheduled at period p, then the binary variable x_{ap} is set to 1. Otherwise we have $x_{ap} = 0$. To ensure that an activity is assigned to consecutive time periods, we also need binary variables s_{ap}, which are set to 1, if activity a starts at period p. Otherwise we have $s_{ap} = 0$. Note that variable s_{ap} is only introduced, if there are at least $d_a - 1$ consecutive time periods available after period p on the same day:

$$x_{ap} - \sum_{\substack{t=p-d_a+1 \\ s_{at} \text{ exists}}}^{p} s_{at} = 0, \quad a \in A,\ p \in P, \tag{1}$$

$$\sum_{s_{ap} \text{ exists}} s_{ap} = 1, \quad a \in A. \tag{2}$$

Equalities (1) ensure that each activity is assigned to a set of consecutive time periods that all belong to the same day. Equalities (2) guarantee that each activity has exactly one start time period.

One of our main goals for the UCL Timetabling Problem is to minimize the total number of rooms required. While this goal is clearly part of the objective function of the Second Stage, we also need to consider it during the First Stage. We propose the following constraints in order to restrict the total number of activities scheduled per time period:

$$\sum_{a \in A} x_{ap} \leq M, \qquad p \in P, \tag{3}$$

where M is an integer variable that is multiplied by a penalty term p_M in the objective function of the First Stage. Without inequalities (3) arbitrarily many activities could be assigned to the same time period during the First Stage, which could leave us with no possibility to minimize the number of rooms required in the Second Stage.

Second Stage: After solving the First Stage, in the Second Stage we determine feasible rooms for the activities, where we aim to minimize the following objectives:

(a) The number of students, which have no seat during an activity.
(b) The number of empty seats in a room during an activity.
(c) The total number of rooms.

In order to build an ILP model for the Second Stage, we introduce binary variables u_r, y_{ar} and z_{arp} with the following interpretations:

- $u_r = 1$, if at least one activity is scheduled in room r. Otherwise $u_r = 0$.
- $y_{ar} = 1$, if activity a is scheduled in room r. Otherwise $y_{ar} = 0$.
- $z_{arp} = 1$, if activity a is scheduled in room r at period p. Otherwise $z_{arp} = 0$.

Note that the variables y_{ar} and z_{arp} are only introduced, if it is feasible to schedule activity a in room r at period p, i.e. if the activity type matches with the room type, if the activity is assigned to period p in the First Stage and if the room is available at period p. Accordingly we define $P(a)$, $P(r)$ and $P(a, r)$ as the sets of available time periods for activity a, for room r and for their combination respectively. Analogously we specify $A(r)$ and $A(p, r)$ as the sets of feasible activities for room r at period p and $R(a, p)$ as the set of feasible rooms for activity a at period p.

For each feasible activity-room combination we introduce a penalty parameter p_{ar} that gives the absolute value of the difference between the available seats in room r and the number of students registered for activity a. We also introduce the penalty parameter p_r giving the costs for using room r. Now we can state our ILP model:

$$\min \sum_{a \in A,\ r \in R(a,p)} p_{ar} y_{ar} + \sum_{r \in R} p_r u_r \tag{4a}$$

$$\text{s.t.} \sum_{r \in R(a)} z_{arp} = 1, \quad a \in A,\ p \in P(a), \tag{4b}$$

$$d_a y_{ar} - \sum_{p \in P(a,r)} z_{arp} = 0, \quad r \in R,\ a \in A(r), \tag{4c}$$

$$\sum_{a \in A(p,r)} z_{arp} - u_r \leq 0, \quad r \in R,\ p \in P(r), \tag{4d}$$

$$u_r \in \{0, 1\},\ y_{ar} \in \{0, 1\},\ z_{arp} \in \{0, 1\}, \quad r \in R,\ a \in A(r),\ p \in P(a, r). \tag{4e}$$

Equalities (4b) guarantee that exactly one room is assigned to an activity at each time period. Equalities (4c) ensure that the same room is assigned to all time periods of

an activity. Constraints (4d) guarantee that at most one activity is assigned to room r at each time period and also ensure $u_r = 1$, if at least one activity is scheduled in r.

4 Metrics and Results

In order to evaluate the quality of a timetable and its usefulness in practice, it is necessary to introduce a wide range of metrics. In this paper we present a short selection of metrics that were deemed most important by UCL.

Space utilization: The UK's most important metric for determining how well universities use their facilities is space utilization s_u, which is defined as the sum of room frequency and average room occupancy. For computing the room frequency r_f we divide the number of time periods occupied by the number of time periods that are available in rooms, in which at least one activity is scheduled.

Next let us define the room occupancy $o_{a,r}$ of an activity a that is scheduled in room r: $o_{a,r} = \min\left(1, s_a/c_r\right),\ r \in R,\ a \in T(r)$, where s_a denotes the number of students registered for activity a, c_r gives the capacity of room r and the set $T(r)$ contains all activities scheduled in room r. Now the average room occupancy $\bar{r}_o$ is simply the mean of all room occupancies of the considered timetable.

Note that the objective function of the Second Stage of our ILP approach is tailored to minimize both space utilization and the number of students without seats.

Timetable regularity: Timetable regularity measures the consistency of time and room assignments of a timetable throughout the term, assuming that each week of the term has slightly different activities. We count the different start times $s(a, w)$ and room assignments $r(a, w)$ of an activity a in week $w \in W$ via the following function:

$$g(w_1, w_2, \overline{a}) = \begin{cases} 2, & \text{if } s(\overline{a}, w_1) \neq s(\overline{a}, w_2) \text{ and } r(\overline{a}, w_1) \neq r(\overline{a}, w_2), \\ 1, & \text{if either } s(\overline{a}, w_1) \neq s(\overline{a}, w_2) \text{ or } r(\overline{a}, w_1) \neq r(\overline{a}, w_2), \\ 0, & \text{otherwise,} \end{cases}$$

with $w_1, w_2 \in W$, $i \in I$ and $\overline{a} \in A(i, w_1, w_2)$ is an activity that has to be scheduled both in w_1 and w_2 in curriculum i. There are b different combinations of activities, pairs of weeks and curricula and the total number of students is given by $h_t = \sum_{i \in I} h_i$, where $h_i,\ i \in I$, is the number of students registered in curriculum i. Now the timetable regularity TR can be defined as:

$$\mathrm{TR} = 1 - \left(\sum_{i \in I} \left(\sum_{\substack{w_1, w_2 \in W \\ w_1 \neq w_2}} \left(\sum_{\overline{a} \in A(i, w_1, w_2)} g(w_1, w_2, \overline{a}) \right) \right) \cdot h_i \right) / \left(2 \cdot b \cdot h_t\right).$$

We try to maximize *TR* by first including activities that have to be scheduled in most weeks of the term in a base week that is solved with the two stage approach described in the previous section. Afterwards we solve an ILP for each particular week, where we maximize similarity to the base week by adding respective soft constraints.

Computational experiments: Finally we present the results obtained by using our ILP approach on a selection of the original set of UCL curricula, available at http://tinyurl.com/timetabling-lib. All experiments were performed on a Linux 64-bit machine equipped with 4 × Intel(R) Xeon(R) CPU e5-2630 v3@2.40 GHz and 16 GB RAM. We use Gurobi 6.5.1 as our ILP-solver.

Our benchmark set consists of around 250 activities per week with an average length of ≈3.5 time periods. Each week consists of 5 days with 18 time periods (a 30 min) per day. In each week we use around 20 of the available 279 rooms.

We obtained timetables for the whole term within 190 s computing time. The corresponding metrics are:

$$\text{(a) } r_f = 0.49, \quad \text{(b) } \bar{r}_o = 0.78, \quad \text{(c) } s_u = 1.27, \quad \text{(d) } TR = 0.8723.$$

The very high timetable regularity of 87.23% is very important for the 35615 UCL students. With our timetables determined they do not have to adapt to frequent weekly timetable changes. Furthermore the high average occupancy metric shows that on average, used rooms are more than $\frac{3}{4}$ full, which ensures an efficient facility usage. Finally the room frequency metric indicates that rooms, which are used at least once, are occupied almost 50% of the total available time periods.

5 Conclusion

In this paper we presented a solution to the curriculum-based timetabling problem of a real-world institution, the University College London, whose requirements and specifications considerably exceed those of the ITC-2007 problem formulation. Due to space restrictions, we selected only the most significant new problem features and the most interesting metrics for this paper. An extended version of this publication will include the solution to the complete problem with 1000 activities per week and 279 rooms, as well as the full set of modeling features, requirements and metrics.

References

1. Asín Achá, R., Nieuwenhuis, R.: Curriculum-based course timetabling with sat and maxsat. Ann. Oper. Res. **218**(1), 71–91 (2014)
2. Lach, G., Lübbecke, M.E.: Curriculum based course timetabling: new solutions to udine benchmark instances. Ann. Oper. Res. **194**(1), 255–272 (2012)
3. Lü, Z., Hao, J.-K.: Adaptive tabu search for course timetabling. Eur. J. Oper. Res. **200**(1), 235–244 (2010)

4. Lewis, R.: A survey of metaheuristic-based techniques for university timetabling problems. OR Spectr. **30**(1), 167–190 (2008)
5. McCollum, B., Schaerf, A., Paechter, B., McMullan, P., Lewis, R., Parkes, A.J., Gaspero, L.D., Qu, R., Burke, E.K.: Setting the research agenda in automated timetabling: The second international timetabling competition. INFORMS J. Comput. **22**(1), 120–130 (2010)
6. Zhang, L., Lau, S.: Constructing university timetable using constraint satisfaction programming approach. In: CIMCA-IAWTIC'06, vol. 02, pp. 55–60. IEEE Computer Society (2005)

Creating Worst-Case Instances for Lower Bounds of the 2D Strip Packing Problem

Torsten Buchwald and Guntram Scheithauer

Abstract We present a new approach to create instances with high absolute worst-case performance ratio of common lower bounds for the two-dimensional rectangular Strip Packing Problem. The idea of this new approach is to optimize the width and the height of all items regarding the absolute worst case performance ratio of the lower bound. Therefore, we model the pattern related to the lower bound as a solution of an ILP problem and merge this model with the Padberg-type model of the two-dimensional Strip Packing Problem. The merged model maximizes the absolute worst-case performance ratio of the lower bound. We introduce this new model for the horizontal bar relaxation and the horizontal contiguous bar relaxation.

1 Introduction

In this paper, we consider the two-dimensional Strip Packing Problem (SPP) with rectangular items. Let a set $I := \{1, \dots, n\}$ of non-rotatable rectangles R_i (items) of width $w_i \leq 1$ and height $h_i \leq 1$ be given. The items have to be packed into a strip of width 1 and minimal height OPT such that the items do not overlap each other.

A lot of lower bounds are known for this problem, but for most of them the exact absolute worst-case performance ratio, which is the supremum over all instances of the fraction of the optimal value and the lower bound, is unknown. To reduce the gap between a proven upper bound of the absolute worst-case performance ratio and the performance ratio of an instance having maximal ratio known so far, it is necessary to decrease the theoretical upper bound or to find instances with greater performance ratio.

T. Buchwald (✉) · G. Scheithauer
Institute of Numerical Mathematics, Dresden University of Technology, Dresden, Germany
e-mail: torsten.buchwald@tu-dresden.de

G. Scheithauer
e-mail: guntram.scheithauer@tu-dresden.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_15

In the following, we introduce a new approach to compute such worst-case instances. For several lower bounds, we show how to model this issue as an optimization problem which maximizes the performance ratio within a subset of SPP instances. In this way, we obtain the absolute worst-case performance ratio of these lower bounds for the considered subsets.

2 Modeling Lower Bounds for the SPP

In this paper, we consider two lower bounds: (binary) horizontal bar relaxation and contiguous (binary) horizontal bar relaxation [1]. We show how the optimization problems addressed above can be modeled. Since we aim to maximize the absolute worst-case performance ratio which is a fraction, we linearize this objective function by fixing the optimal value and minimizing the height of the lower bound. For our models we assume that all items can be packed using a strip height of at most 1, i.e., $OPT \leq 1$ holds. To ensure that this condition is fulfilled, our model contains a Padberg-type model [2]. The second part of our model describes the considered lower bound. Hence, we aim to minimize the height of the lower bound solution depending on the widths and heights of the items.

2.1 Modeling an Optimal Pattern

The main issue of our approach is the managing of the optimal value of the SPP instance which results by minimization and which should be maximized in comparison to the lower bound at the same time. On the one hand, by definition of the SPP, we search for a feasible pattern with minimal value but we also try to get an instance with lower bound as small as possible. So, in order to get a large absolute worst-case performance ratio the optimal value should be maximized with respect to the considered instance set. To resolve this issue, we fix the optimal value and iterate over all possible patterns to provide the optimal solution, that means, the height of the considered feasible pattern has to be 1 and all other patterns are either infeasible or have height at least 1. Obviously, any particular pattern can be characterized by the relative positions of each pair of items (left, right, above, below). Regarding symmetry and permutation aspects, we have only 4 different patterns with $n = 3$ items (Fig. 1).

The pattern which should be the optimal pattern is called *current pattern*. For the current pattern we have to guarantee that this pattern is feasible and has height of at least 1. To model the feasibility of a current pattern, we consider maximal subsets of items which are placed next to each other in this pattern. These subsets are called *horizontal slices* of the considered pattern. Obviously, the current pattern is only feasible if the total width of each horizontal slice of this pattern does not exceed the width of the strip. Analogously to the horizontal slices, we denote the

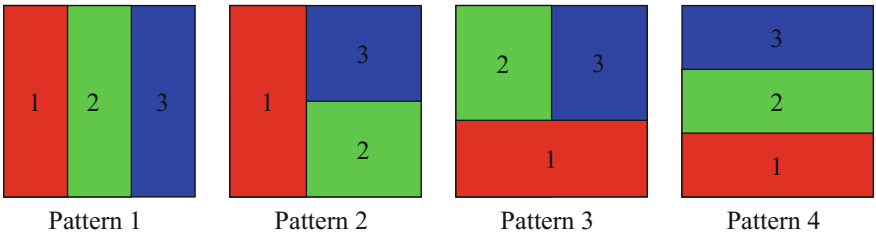


Fig. 1 Possible patterns of 3 items

maximal subsets of items which are placed above each other in the considered pattern as vertical slices. To ensure that the current pattern has height of at least 1, we have to enforce that the total height of at least one vertical slice of this pattern is at least 1.

To model these conditions, we introduce two binary variables x_A and y_A for each possible subset $A \in \mathscr{P}_n$ of items where $\mathscr{P}_n$ denotes the set of all subsets of $\{1, \ldots, n\}$ with at least two elements. (Note that singletons are not meaningful in our approach.) These variables belong to the horizontal and vertical slices which coincide with the corresponding subsets of items. Since x_A and y_A indicate whether the total width or the total height of item set A exceeds the respective boundary of the strip, we apply the following inequalities with a small $\varepsilon > 0$:

$$(1 - x_A) * W + \sum_{i \in A} w_i \geq W + \varepsilon \quad \text{for all } A \in \mathscr{P}_n, \tag{1}$$

$$(1 - y_A) * 1 + \sum_{i \in A} h_i \geq 1 \quad \text{for all } A \in \mathscr{P}_n. \tag{2}$$

In the first inequality, if $x_A = 1$, then the total width of the items of A has to exceed the strip width, otherwise the horizontal slice can be feasible or not. Analogously, if $y_A = 1$, the second inequality ensures that the total height of the items of A is at least 1. Let CP denote the current pattern. Furthermore, let $HS(P)$ and $VS(P)$ denote the set of horizontal and vertical slices of a pattern P, respectively. Then we model the feasibility of CP by demanding the following inequalities:

$$\sum_{i \in A} w_i \leq W \quad \text{for all } A \in HS(CP) \cap \mathscr{P}_n. \tag{3}$$

Note that these inequalities imply

$$x_A = 0 \quad \text{for all } A \in HS(CP) \cap \mathscr{P}_n.$$

Moreover we ensure that the height of the current pattern is at least 1 by using the following inequality:

$$\sum_{A \in VS(CP) \cap \mathscr{P}_n} y_A \geq 1. \tag{4}$$

Note that this inequality leads to an infeasible model for that pattern, where all items are packed next to each other (pattern 1 in Fig. 1). But this is not relevant, since the optimal value of the lower bound is equal to 1 in this case. Summarizing the usage of all inequalities (1)–(4) within the model ensures that the current pattern is feasible and has height of at least 1. However, We still have to guarantee that no other pattern has a height less than 1.

2.2 Other Patterns

To ensure that no other feasible pattern gives a better solution than the current pattern CP, we need to add appropriate inequalities for each other pattern. Let OP be any other pattern. Then, to guarantee that pattern OP is not a better feasible pattern than CP, we have to enforce that either OP is infeasible or that it requires a strip height of at least 1. This is modeled by the following inequality:

$$\sum_{A \in HS(OP) \cap \mathscr{P}_n} x_A + \sum_{B \in VS(OP) \cap \mathscr{P}_n} y_B \geq 1. \tag{5}$$

Adding this inequality for each other pattern we ensure that the current pattern becomes an optimal pattern. Since the height of CP is enforced to be at least 1, now we can minimize the value of the considered relaxation in order to find an instance with a maximal absolute worst case performance ratio.

Since the whole problem is too complex, we consider particular subsets of instances defined as follows: The maximum number of original rectangular items of size $w_i \times h_i$ in the instance is restricted by a given number N. Clearly, the variables w_i and h_i, $i \in I = \{1, \ldots, N\}$, have to fulfill the constraints

$$\varepsilon \leq w_i \leq 1, \quad i \in I, \tag{6}$$
$$0 \leq h_i \leq 1, \quad i \in I. \tag{7}$$

Let (x_i, y_i), $i \in I$, denote the allocation point (lower left corner) of item i, and let u_{ij} and v_{ij}, $i,j \in I$, $i \neq j$, be binary variables (according to [2]) to characterize the mutual position of items i and j in the pattern, then the feasibility of the instance is enforced by

$$0 \leq x_i \leq 1 - w_i, \quad i \in I, \tag{8}$$
$$0 \leq y_i \leq 1 - h_i, \quad i \in I, \tag{9}$$
$$x_i + w_i \leq x_j + 1 - u_{ij}, \quad i,j \in I, \quad i \neq j, \tag{10}$$
$$y_i + h_i \leq y_j + 1 - v_{ij}, \quad i,j \in I, \quad i \neq j, \tag{11}$$
$$u_{ij} + u_{ji} + v_{ij} + v_{ji} = 1, \quad i,j \in I, \quad i \neq j. \tag{12}$$

According to the horizontal (contiguous) bar relaxation, any item is partitioned into *s item parts* of size $w_i \times h_{ik}$, $k \in K = \{1, \ldots, s\}$, by horizontal cuts. Naturally, we have the constraints

$$0 \le h_{ik}, \quad i \in I, \quad k \in K, \tag{13}$$

$$\sum_{k=1}^{s} h_{ik} = h_i, \quad i \in I. \tag{14}$$

To model the feasibility of the solution related to the bound, let (x_{ik}, y_{ik}), $i \in I, k \in K$, denote the allocation point of item part (i,k). Moreover, let u_{ikjl} and v_{ikjl}, $i,j \in I$, $k,l \in K$, $i \ne j$, be binary variables to characterize the mutual position of item parts (i,k) and (j,l) in the pattern of the bar relaxation. To guarantee that the item parts can be packed into the strip, using an minimal height H, we add the following constraint:

$$y_{ik} + h_{ik} \le y_{i,k+1}, \quad i \in I, \quad k = 1, \ldots, s-1, \tag{15}$$

$$y_{is} + h_{is} \le H, \quad i \in I, \tag{16}$$

$$0 \le x_{ik} \le 1 - w_i, \quad i \in I, \quad k \in K, \tag{17}$$

$$x_{ik} + w_i \le x_{jl} + 1 - u_{ikjl}, \quad i,j \in I, \quad k,l \in K, \quad i \ne j, \tag{18}$$

$$y_{ik} + h_{ik} \le y_{jl} + 1 - v_{ikjl}, \quad i,j \in I, \quad k,l \in K, \quad (i,k) \ne (j,l), \tag{19}$$

$$u_{ikjl} + u_{jlik} + v_{ikjl} + v_{jlik} = 1, \quad i,j \in I, \quad k,l \in K, \quad i \ne j. \tag{20}$$

Summarizing, the whole model to compute an instance with maximal worst-case performance ratio is as follows:

$$H \to \min$$

subject to constraints

1. (6)–(7) for the size parameters of the items,
2. (8)–(12) for the feasibility of the instance computed,
3. (13)–(20) for the partition of items and feasible packability of all item parts,
4. (1)–(5) for the optimality of the (current) pattern and optimal height 1.

To extend the model to the horizontal contiguous bar relaxation, we just need to replace constraints (15) by

$$y_{ik} + h_{ik} = y_{i,k+1}, \quad i \in I, \quad k = 1, \ldots, s-1, \tag{15 $*$}$$

which ensure that the solution of the relaxation fulfills the contiguous property.

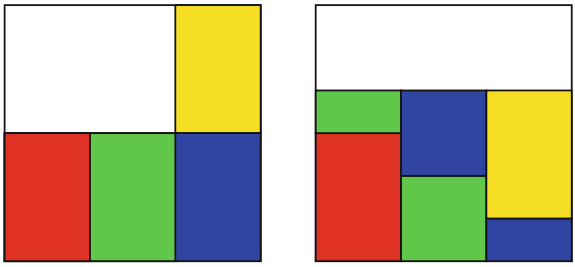


Fig. 2 Instance with maximal worst-case performance ratio for $N = 4$ and $s = 2$

3 Preliminary Results

Up to now, we have only solved the model with a given small number N of items which are partitioned each into $s = 2$ item parts. For the horizontal bar relaxation we observed, that the patterns of the solutions have the same structure, which is displayed in Fig. 2 for the case $N = 4$.

The height of the solution of the bar relaxation is always equal to $N/(2N - 2)$ which asymptotically proves that the absolute worst case performance ratio of the horizontal bar relaxation is at least 2.

We also applied the model for the contiguous horizontal bar relaxation for only small N and s. But up to now all solutions obtained provide lower bound 1, and therefore, ratio 1. For larger n, we will get absolute worst-case performance ratios larger than 1, due to [1]

4 Conclusions and Outlook

In this paper, we proposed a new approach to obtain worst-case instances for the two-dimensional Strip Packing Problem when the number of items and item parts is limited. We implemented this approach for two lower bounds and presented first promising results for the horizontal bar relaxation. We are optimistic that we will obtain similar results for the contiguous horizontal bar relaxation.

It will be part of our future work to apply this approach to other lower bounds. Moreover, we will try to improve the model by further examining the structure of the problem and optimizing the performance of our approach with respect to this structure.

References

1. Belov, G., Kartak, V., Rohling, H., Scheithauer, G.: One-dimensional relaxations and LP bounds for orthogonal packing. Int. Trans. Oper. Res. **16**, 745–766 (2009)
2. Padberg, M.: Packing small boxes into a big box. Math. Meth. OR **1**, 1–21 (2000)

Low-Rank/Sparse-Inverse Decomposition via Woodbury

Victor K. Fuentes and Jon Lee

Abstract Based on the Woodbury matrix identity, we present a heuristic and a test-problem generation method for decomposing an invertible input matrix into a low-rank component and a component having a sparse inverse.

1 Introduction

Our starting point is the well-known low-rank/sparse decomposition problem

$$\min \left\{ \bar{\tau}\|A\|_0 + (1-\bar{\tau})\text{rank}(B) \ : \ A + B = \bar{C} \right\}, \qquad (\mathscr{D}_0)$$

where $\bar{C}$ is an $m \times n$ input matrix, $0 < \bar{\tau} < 1$, $\|\cdot\|_0$ counts the number of non-zeros, and A and B are matrix variables (see [2]). The problem $\mathscr{D}_0$ is a central problem in the area of statistical model selection, where the sparse matrix can correspond to a Gaussian graphical model, and the low-rank matrix can capture the effect of latent, unobserved random variables.

1.1 Convex Approximation

The problem $\mathscr{D}_0$ is ordinarily approached by using the (element-wise) 1-norm as an approximation of $\|\cdot\|_0$ and using the nuclear norm (sum of the singular values) as an approximation of rank. So we are led to the approximation

V.K. Fuentes: Supported in part by MICDE.
J. Lee: Supported by NSF grant CMMI-1160915 and ONR grant N00014-14-1-0315.

V.K. Fuentes (✉) · J. Lee
University of Michigan, Ann Arbor, MI, USA
e-mail: vicfuen@umich.edu

J. Lee
e-mail: jonxlee@umich.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_16

$$\min \left\{ \bar{\tau}\|A\|_1 + (1-\bar{\tau})\|B\|_* \right) \ : \ A + B = \bar{C} \Big\}, \qquad (\mathscr{D}_1)$$

where $\|A\|_1 := \sum_{i,j} |a_{ij}|$ and $\|B\|_*$ denotes the sum of the singular values of B. This approach has some very nice features. First of all, because we have genuine norms now in the objective function, this approximation $\mathscr{D}_1$ is a convex optimization problem, and so we can focus our attention on seeking a local optimum of $\mathscr{D}_1$ which will then be a global optimum of $\mathscr{D}_1$. Still, the objective function of $\mathscr{D}_1$ is not differentiable everywhere, and so we are not really out of the woods. However, the approximation $\mathscr{D}_1$ can be re-cast (see [2, Appendix A]) as a semidefinite-optimization problem

$$\begin{aligned} &\min \ \bar{\tau}\mathbf{e}'S\mathbf{e} + (1-\bar{\tau})\frac{1}{2}\left(\mathrm{tr}(W_1) + \mathrm{tr}(W_2)\right) \\ &A + B = \bar{C}, \qquad -S \le A \le S, \qquad \begin{pmatrix} W_1 & B \\ B' & W_2 \end{pmatrix} \succeq 0, \end{aligned}$$

which is efficiently solvable in principle (see [7]). We note that semidefinite-optimization algorithms are not at this point very scalable. Nonetheless, there are first-order methods for this problem that do scale well and allow us to quickly get good approximate solutions for large instances (see [1]).

Also, it is interesting to note that to solve $\mathscr{D}_0$ globally (with additional natural constraints bounding the feasible region), a genuine relaxation of $\mathscr{D}_0$ closely related to $\mathscr{D}_1$ should be employed (see [5]).

1.2 Generating Test Problems via the Recovery Theory

Another extremely nice feature of the convex approximation $\mathscr{D}_1$ is a "recovery theory". Loosely speaking it says the following: If we start with a sparse matrix $\bar{A}$ that does not have low rank, and a low-rank matrix $\bar{B}$ that is not sparse, then there is a non-empty interval $\mathscr{I} := [\bar{\tau}_\ell, \bar{\tau}_u] \subset [0,1]$ so that for all $\bar{\tau} \in \mathscr{I}$, the solution of the approximation $\mathscr{D}_1$ is uniquely $A = \bar{A}$ and $B = \bar{B}$.

The recovery theory suggests a practical paradigm for generating test problems for algorithms for $\mathscr{D}_1$.

Procedure 1

1. Generate a random sparse matrix $\bar{A}$ that with high probability will not have low rank. For example, for some natural number $\ell \ll \min\{m,n\}$, randomly choose $\ell \cdot \min\{m,n\}$ entries of $\bar{A}$ to be non-zero, and give those entries values independently chosen from some continuous distribution.
2. Generate a random low-rank matrix $\bar{B}$ that with high probability will not be sparse. For example, for some natural number $k \ll \min\{m,n\}$, make an $m \times k$ matrix $\bar{U}$ and a $k \times n$ matrix $\bar{V}$, with entries chosen independently from some continuous distribution, and let $\bar{B} := \bar{U}\bar{V}$.
3. Let $\bar{C} := \bar{A} + \bar{B}$.

4. Perform a search on $[0, 1]$ to find a value $\bar{\tau}^*$ so that the solution of $\mathscr{D}_1$ with $\bar{\tau} = \bar{\tau}^*$ is $A = \bar{A}$ and $B = \bar{B}$.
5. Output: $\bar{C}$, $\bar{\tau}^*$.

The recovery theory tells us that there will be a value of $\bar{\tau}^*$ for which the solution of $\mathscr{D}_1$ with $\bar{\tau} = \bar{\tau}^*$ is $A = \bar{A}$ and $B = \bar{B}$. What is not completely clear is that there is a disciplined manner of searching for such a $\bar{\tau}^*$ (step 4). Let $A_{\bar{\tau}}, B_{\bar{\tau}}$ be a solution of $\mathscr{D}_1$, with the notation emphasizing the dependence on $\bar{\tau}$. We define the univariate function

$$f(\bar{\tau}) := \|\bar{A} - A_{\bar{\tau}}\|_F = \|\bar{B} - B_{\bar{\tau}}\|_F = \frac{1}{2}\left(\|\bar{A} - A_{\bar{\tau}}\|_F + \|\bar{B} - B_{\bar{\tau}}\|_F\right).$$

Clearly, for $\bar{\tau} = 0$, the solution of $\mathscr{D}_1$ will be $B = 0$, $A = \bar{C}$ and $f(0) = \|\bar{B}\|_F$. Likewise, for $\bar{\tau} = 1$, the solution of $\mathscr{D}_1$ will be $A = 0$, $B = \bar{C}$ and $f(1) = \|\bar{A}\|_F$. For $\bar{\tau}^* \in \mathscr{I}$, f is minimized with $f(\bar{\tau}^*) = 0$, And we can hope that f is quasiconvex and we may quickly find a minimizer via a bisection search.

2 Low-Rank/Sparse-Inverse Decomposition

Now, we turn our attention to a closely related problem—which is our main focus. We assume now that $\bar{G}$ is an order-*n square* input matrix, $0 < \bar{\tau} < 1$, and our goal is to solve the low-rank/sparse-*inverse* decomposition problem:

$$\min\left\{\bar{\tau}\|E^{-1}\|_0 + (1 - \bar{\tau})\text{rank}(F) \;:\; E + F = \bar{G}\right\}. \qquad (\mathscr{H}_0)$$

Note that, generally, it may be that $\bar{G}$ is not invertible, but in the approach that we present here, we will assume that $\bar{G}$ is invertible.

The problem $\mathscr{H}_0$ can capture an interesting problem in statistics. In that setting, $\bar{G}$ can be a sample covariance matrix. Then E can be the true covariance matrix that we wish to recover. In some settings, the inverse of E (known as the *precision matrix*) can be of unknown sparse structure—a zero entry in the inverse of E identifies when a pair of variables are conditionally (on the other $n - 2$ variables) independent. We do note that for this application, because the sample covariance matrix and the true covariance matrix are positive semidefinite, there are alternative approaches, based on convex approximations, that are very attractive (see [6] and the references therein). So our approach can best be seen as having its main strength for applications in which $\bar{G}$ is not positive semidefinite.

2.1 An Algorithmic Approach via the Woodbury Identity

The algorithmic approach that we take is as follows.

Procedure 2

1. Let $\bar{C} := \bar{G}^{-1}$.
2. Apply *any* approximation method for $\mathscr{D}_0$, yielding some A (and B). For example, we can solve $\mathscr{D}_1$.
3. Output $E := A^{-1}$ and $F := \bar{G} - E$.

Our methodology is justified by the *Woodbury matrix identity* (see [4]). In step 2, we find a decomposition $A + B = \bar{G}^{-1}$, with A sparse and B low rank. Now, suppose that $\text{rank}(B) = k$. Then it can be written as $B = UV$, where U is $n \times k$ and V is $k \times n$. By the Woodbury identity, we have

$$\bar{G} = \bar{C}^{-1} = (A + B)^{-1} = (A + UV)^{-1} = A^{-1} - A^{-1}U(I + VA^{-1}U)^{-1}VA^{-1}.$$

Because A is sparse, we have that E^{-1} is sparse. Finally, we have $F = -A^{-1}U(I + VA^{-1}U)^{-1}VA^{-1}$ which has rank no more than k.

2.2 Generating Test Problems Without a Recovery Theory

We could try to work with the approximation

$$\min \left\{ \bar{\tau}\|E^{-1}\|_1 + (1 - \bar{\tau})\|F\|_* \right) \ : \ E + F = \bar{G} \Big\} \qquad (\mathscr{H}_1)$$

of $\mathscr{H}_0$, but $\mathscr{H}_1$ is not a convex optimization problem, and there is no direct recovery theory for it. But we can exploit the correspondence (via the Woodbury identity) with $\mathscr{D}_1$ to generate test problems for the non-convex problem $\mathscr{H}_1$. In analogy with Procedure 1 of Sect. 1.2, we employ the following methodology, which incorporates our heuristic Procedure 2.

Procedure 3

1. Generate a random sparse square invertible matrix $\bar{A}$. This may have to be done with a few trials to ensure that $\bar{A}$ is invertible. Let $\bar{E} := \bar{A}^{-1}$.
2. Generate a random low-rank square matrix $\bar{B} := \bar{U}\bar{V}$ that with high probability is not sparse, as described in step 2 of Procedure 1.
3. Let $\bar{F} := -\bar{A}^{-1}\bar{U}(I + \bar{V}\bar{A}^{-1}\bar{U})^{-1}\bar{V}\bar{A}^{-1}$, and let $\bar{G} := \bar{E} + \bar{F}$.
4. Let $\bar{C} := \bar{G}^{-1}$. Search on $[0, 1]$ to find a $\bar{\tau}^*$ seeking to minimize $f(\bar{\tau}) := \|\bar{A} - A_{\bar{\tau}}\|_F = \|\bar{B} - B_{\bar{\tau}}\|_F$.
5. Output: $\bar{G}$, $\bar{\tau}^*$.

Because of the way we have engineered $\bar{F}$ in Procedure 3, we take advantage of the ordinary recovery theory for $\mathscr{D}_1$.

3 Computational Experiments

We carried out some preliminary computational experiments for Procedure 3, using $n = 75$. We did six experiments, one each with the rank of $\bar{B}$ at $k = 3, 6, 9, 12, 15, 18$. For each value of k, we chose $\bar{A}^{-1}$ to have $(k + 1)n$ non-zeros. So, as k increases, the rank of $\bar{B}$ is increasing and the number of non-zeros in $\bar{A}^{-1}$ is increasing. Therefore, we can expect that as k increases, the "window of recovery" (i.e., the set of $\bar{\tau}$ so that $f(\bar{\tau}) = 0$) gets smaller and perhaps vanishes; and once it vanishes, we can expect that the minimum value of $f(\bar{\tau})$ is increasing with k. We can see that this is all borne out in

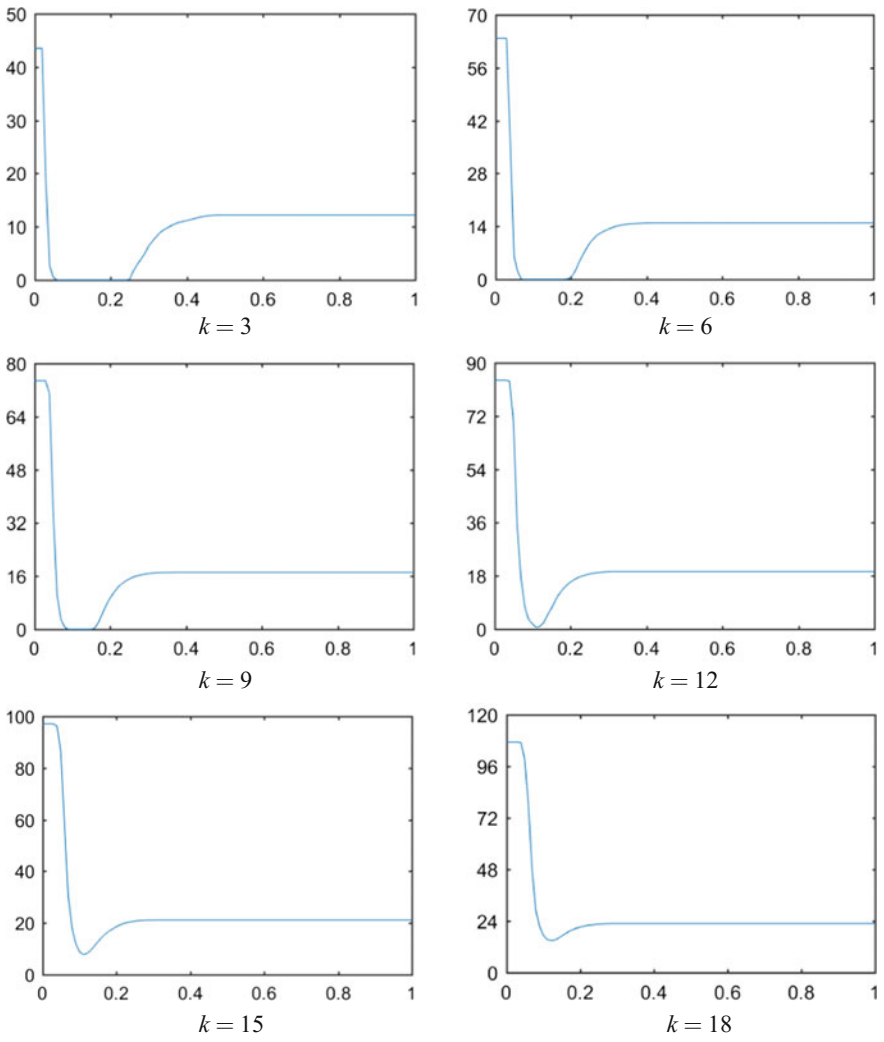
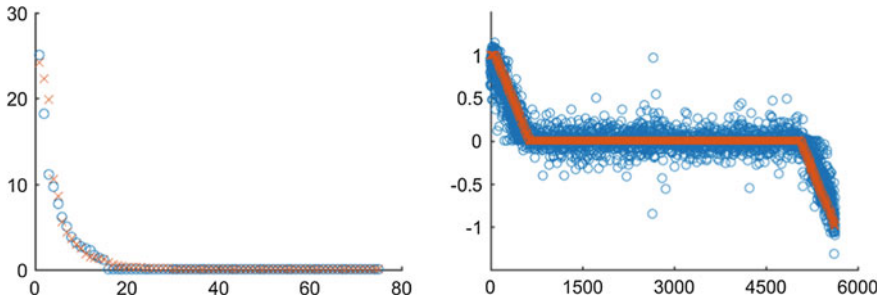


Fig. 1 $f(\bar{\tau})$ versus $\bar{\tau}$ $(n = 75)$



Singular values: input $\bar{F}$; output $\bar{G} - A_{\bar{\tau}}^{-1}$ Entries: sorted input $\bar{E}^{-1}$ vs. output $A_{\bar{\tau}}$

Fig. 2 $n = 75,\ k = 15$

Fig. 1. Next, we focus on the $k = 15$ case, where the minimum of $f(\bar{\tau})$ is substantially above 0. Even in such as case, we can see in Fig. 2 that there is substantial recovery, attesting to the efficacy of our heuristic Procedure 2.

4 Conclusions

We presented a heuristic and a means of generating test problems for the low-rank/sparse-inverse decomposition problem on invertible input. Our method can also be used for generating a starting point for local optimization of $\mathscr{H}_1$.

We are presently working on a new approach to $\mathscr{H}_0$ based on a convex relaxation of $\mathscr{H}_1$. This new approach is much more computationally intensive than the method that we presented here, which we leverage for validating our new approach. Our new approach does not require that the input matrix be invertible. In fact, it can equally apply to even-more-general low-rank/sparse-*pseudoinverse* decomposition problems (see [3]).

References

1. Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. Found. Trends Mach. Learn. **3**(1), 1–122 (2010)
2. Chandrasekaran, V., Sanghave, S., Parrilo, P.A., Willsky, A.S.: Rank-sparsity incoherence for matrix decomposition. SIAM J. Optim. **21**(2), 572–596 (2011)
3. Fuentes, V.K., Fampa, M., Lee, J.: Sparse pseudoinverses via LP and SDP relaxations of Moore-Penrose. CLAIO 343–350 (2016)
4. Hager, W.W.: Updating the inverse of a matrix. SIAM Rev. **31**(2), 221–239 (1989)
5. Lee, J., Zou, B.: Optimal rank-sparsity decomposition. J. Glob. Optim. **60**(2), 307–315 (2014)

6. Scheinberg, K., Ma, S., Goldfarb, D.: Sparse inverse covariance selection via alternating linearization methods. NIPS 2101–2109 (2010)
7. Wolkowicz, H., Saigal, R., Vandenberghe, L.: Handbook of Semidefinite Programming: Theory, Algorithms, and Applications. Kluwer, London (2000)

On-Line Algorithms for Controlling Palletizers

Frank Gurski, Jochen Rethmann and Egon Wanke

Abstract We consider the FIFO STACK-UP problem which arises in delivery industry, where bins have to be stacked-up from conveyor belts onto pallets. Given are k sequences $q_1, \ldots, q_k$ of labeled bins and a positive integer p. The goal is to stack-up the bins by iteratively removing the first bin of one of the k sequences and put it onto a pallet located at one of p stack-up places. Each of these pallets has to contain bins of only one label, bins of different labels have to be placed on different pallets. After all bins of one label have been removed from the given sequences, the corresponding stack-up place becomes available for a pallet of bins of another label. In this paper we consider on-line algorithms for instances where we only know the next c bins of every sequence instead of the complete sequences. We implemented our algorithms and could show that for realistic, but randomly generated instances our best approach leads only 12% more stack-up places than an optimal off-line solution. On the other hand we could show worst-case examples which show an arbitrary large competitive factor when comparing our on-line solutions with optimal off-line solutions.

1 Introduction

We consider the combinatorial problem of stacking up bins from conveyor belts onto pallets. This problem originally appears in *stack-up systems* or *palletizing systems* that play an important role in delivery industry and warehouses. Stack-up systems are often the back end of *order-picking systems*. A detailed description of the applied background of such systems is given in [1, 2].

F. Gurski (✉) · E. Wanke
Institute of Computer Science, University of Düsseldorf, 40225 Düsseldorf, Germany
e-mail: frank.gurski@hhu.de

E. Wanke
e-mail: e.wanke@hhu.de

J. Rethmann
Faculty of Electrical Engineering and Computer Science, Niederrhein University of Applied Sciences, 47805 Krefeld, Germany
e-mail: jochen.rethmann@hs-niederrhein.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_17

The bins that have to be stacked-up onto pallets reach the stack-up system on a conveyor belt. At the stack-up system the bins are picked-up by robotic arms or stacker cranes and moved onto pallets. The pallets are located at *stack-up places*. This picking process can be performed in different ways depending on the architecture of the palletizing system. Full pallets are carried away by automated guided vehicles or by another conveyor system, while new empty pallets are placed at free stack-up places.

We consider so-called *multi-line palletizing systems*, where there are several buffer conveyors from which the bins are picked-up. The robotic arms or stacker cranes and the stack-up places are located at the end of these conveyors. We assume that the assignment of the bins to the conveyors and the order of bins within each conveyor is given. If further each arm can only pick-up the first bin of one of the buffer conveyors, then the system is called a *FIFO palletizing system*. Such systems can be modeled by several simple queues. Figure 1 shows a sketch of a simplified stack-up system with 2 buffer conveyors and 3 stack-up places.

In the following we describe a stack-up processing using a simple example. For some technical definitions see [3–5]. Given two sequences $q_1 = (b_1, \ldots, b_4) = [a, b, a, b]$ and $q_2 = (b_5, \ldots, b_{10}) = [c, d, c, d, a, b]$. Each bin b_i is labeled with a *pallet symbol* $plt(b_i)$. Every row of Fig. 2 represents a *configuration* (Q, Q'), where the first entry Q is the initial list of sequences of bins and the second entry $Q' = (q_1^i, q_2^i)$

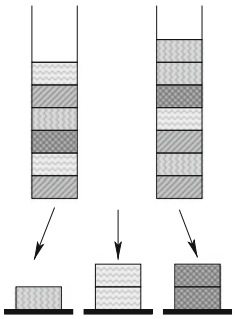


Fig. 1 A FIFO stack-up system

i	q_1^i	q_2^i	$front(Q_i)$	$open(Q, Q_i)$	remove
0	$[a,b,a,b]$	$[\underline{c},d,c,d,a,b]$	$\{a,c\}$	$\emptyset$	b_5
1	$[a,b,a,b]$	$[c,\underline{d},c,d,a,b]$	$\{a,d\}$	$\{c\}$	b_6
2	$[a,b,a,b]$	$[c,d,\underline{c},d,a,b]$	$\{a,c\}$	$\{c,d\}$	b_7
3	$[a,b,a,b]$	$[c,d,c,\underline{d},a,b]$	$\{a,d\}$	$\{d\}$	b_8
4	$[\underline{a},b,a,b]$	$[c,d,c,d,a,b]$	$\{a\}$	$\emptyset$	b_1
5	$[a,b,a,b]$	$[c,d,c,d,\underline{a},b]$	$\{a,b\}$	$\{a\}$	b_9
6	$[a,\underline{b},a,b]$	$[c,d,c,d,a,b]$	$\{b\}$	$\{a\}$	b_2
6	$[a,b,a,b]$	$[c,d,c,d,a,\underline{b}]$	$\{a,b\}$	$\{a,b\}$	b_{10}
7	$[a,b,\underline{a},b]$	$[c,d,c,d,a,b]$	$\{a\}$	$\{a,b\}$	b_3
8	$[a,b,a,\underline{b}]$	$[c,d,c,d,a,b]$	$\{b\}$	$\{b\}$	b_4
9	$[a,b,a,b]$	$[c,d,c,d,a,b]$	$\emptyset$	$\emptyset$	–

Fig. 2 A processing of two given sequences with 2 stack-up places

is the list of sequences that remain to be processed. Already removed bins are shown in greyed out. The transition from one row to the next row, i.e. the removal of the first bin from one subsequence $q' \in Q'$ is called *transformation step*. A pallet t is called *open* in configuration (Q, Q'), if a bin of pallet t is contained in some $q'_i \in Q'$ and if another bin of pallet t is contained in some $q_j - q'_j$ for $q_j \in Q, q'_j \in Q'$. The *set of open pallets* in configuration (Q, Q') is denoted by $open(Q, Q')$. For some configuration (Q, Q') we define $front(Q')$ to be the pallets of the first bins of the queues of Q'. A configuration (Q, Q') is called a *decision configuration*, if the first bin of each sequence $q' \in Q'$ is destined for a non-open pallet, i.e. $front(Q') \cap open(Q, Q') = \emptyset$. For some list of sequences Q we define by $plts(Q)$ the set of all pallets in all sequences of Q.

A sequence of transformation steps that transforms the list Q of k sequences into k empty subsequences is called a *processing* of Q. The order in which the bins are removed from the sequences within a processing of Q is called a *bin solution* of Q and the order in which the pallets are opened during the processing of Q is denoted as *pallet solution*. In Fig. 2 we obtain bin solution $B = (b_5, b_6, b_7, b_8, b_1, b_9, b_2, b_{10}, b_3, b_4)$, i.e. a permutation of the bins, and the pallet solution $T = (c, d, a, b)$.

The FIFO Stack-Up problem is to decide for a given list Q of k sequences and a positive integer p whether there is a processing of Q, such that in each configuration during the processing of Q at most p pallets are open. The FIFO Stack-Up problem is NP-complete even if the number of bins per pallet is bounded, but can be solved in polynomial time if the number k of sequences or the number p of stack-up places is fixed [6]. A dynamic programming solution for the problem is shown in [3]. Parameterized algorithms and a linear programming approach for computing a pallet solution for the problem is given in [5]. A breadth first search solution combined with some cutting technique for the problem was presented in [4]. An experimental study of algorithms for controlling palletizers was given in [7].

2 On-Line Solutions for the FIFO Stack-Up Problem

Within an off-line processing the complete sequences are known in advance, whereas in an on-line processing only a lookahead of the next c bins of each of the subsequences are known in each configuration. Additionally each algorithm knows the number of bins for each pallet, in order to recognize the end of the pallets. This is no restriction, since in practice this number is known from the costumer orders and the distribution of the costumer orders onto bins. On-line algorithms make their decisions based on partial informations of the input.

We use the following variables: k denotes the number of sequences, p stands for the number of stack-up places, m represents the number of pallets, and n denotes the number of bins. Let Q be a list of k sequences and $t \in plts(Q)$ be some pallet. We define by $\#bins(t, Q)$ the number of bins for t in all k sequences of Q. Let (Q, Q') be some decision configuration during the processing of Q by some on-line algorithm, $t \in plts(Q)$ be some pallet, and $c \in \mathbb{N}$. We define $\#_c front(t, Q')$ the number of bins

which are destinated for t on the first c positions in all sequences of Q'. The special case where $c = 1$ is denoted by $\#front(t, Q')$.

In order to open the next pallet $t \in front(Q')$ we define the following rules.

- Round Robin (RR): Choose t such that for the i-th decision configuration the first bin of sequence $q_{((i-1) \bmod k)+1}$ is destinated for pallet t.
- Random Robin (RD): Choose t such that the first bin of a randomly chosen sequence q_i is destinated for pallet t.
- Least Recently (LR): Choose t such that $\min\{i \mid 1 \leq i \leq \ell,\ t \in front(Q_i)\}$ is as small as possible, where $(Q, Q_1), \ldots, (Q, Q_\ell)$ are the decision configurations up to configuration (Q, Q').
- Biggest Front (BF): Choose t such that $\#front(t, Q')$ is as high as possible.
- Most Frequently (MF): Choose t such that $\#_c front(t, Q')$ is as high as possible.
- Most Executed (ME): Choose t such that $\frac{\#_c front(t,Q')}{\#bins(t,Q)}$ is as high as possible.
- Least Leftover (LL): Choose t such that $\#bins(t, Q) - \#_c front(t, Q')$ is as small as possible.
- Early Closure (EC): Now we are interested in bins which are the *last of their pallet* in configuration (Q, Q'), i.e. bins for which there is no further bin of the same pallet symbol in all sequences of Q'. Every such bin is destinated for an open pallet, since we only consider lists of sequences that together contain at least two bins for each pallet. Last bins of their pallet can be determined by the known number of bins for each pallet. For every sequence q_i we compute a score by summing $c - j + 1$ for every position $j = 1, \ldots, c$ on which there is a last bin of its pallet in q_i. After that we choose t from a sequence q_i with largest score. If in none of the sequences there is a last bin of its pallet among the first c positions we choose t by Least Recently (LR).

3 Theoretical Analysis

In order to evaluate on-line algorithms from a theoretical point of view, we study their worst-case performance. This is done by a competitive analysis [8, 9], i.e. we compare the performance of our on-line algorithms with optimal off-line solutions. An on-line algorithm is *d-competitive* if it computes a processing of some list Q with at most $p \cdot d$ stack-up places, if Q can be processed with at most p stack-up places.

In our first example we consider the case $c = 1$. To convert the example for $c > 1$ we increase the first bin of each pallet to its c-fold. Further we consider the list $Q_1 = (q_1, q_2, q_3, q_4)$ of the following $k = 4$ sequences, which also can be generalized.

$$q_1 = [a, e, i, m, \ldots], q_2 = [b, f, j, n, \ldots], q_3 = [c, g, k, o, \ldots]$$
$$q_4 = [d, d, h, h, \ell, \ell, p, p, \ldots, a, b, c, e, f, g, i, j, k, m, n, o, \ldots]$$

Then $T_{opt} = (d, h, \ell, p, \ldots, a, b, c, e, f, g, i, j, k, m, n, o, \ldots)$ is an optimal pallet solution for Q_1 using $p = 1$ stack-up place and Round Robin (RR) leads to pallet solu-

tion $T_{RR} = (a, b, c, d, e, f, g, h, i, j, k, \ell, m, n, o, p, \ldots)$ using $p = m \cdot \frac{k-1}{k} + 1$ stack-up places, if $m \bmod k = 0$. Algorithm Least Recently (LR) leads to the same pallet solution $T_{LR} = T_{RR}$.

Next we consider the list $Q_2 = (q'_1, q''_1, q'_2, q''_2, q'_3, q''_3, q_4)$ of the following $k = 7$ sequences, which also can be generalized.

$$\begin{aligned} q'_1 &= [a, e, i, m, \ldots], q''_1 = [a, e, i, m, \ldots] \\ q'_2 &= [b, f, j, n, \ldots], q''_2 = [b, f, j, n, \ldots] \\ q'_3 &= [c, g, k, o, \ldots], q''_3 = [c, g, k, o, \ldots] \\ q_4 &= [d, d, h, h, \ell, \ell, p, p, \ldots, a, b, c, e, f, g, i, j, k, m, n, o, \ldots] \end{aligned}$$

An optimal pallet solution for Q_2 is T_{opt} given above using $p = 1$ stack-up place. Biggest Front (BF) applied on Q_2 leads to pallet solution $T_{BF} = T_{RR}$. Most Frequently (MF) works for $c = 1$ in the same way as BF. For $c > 1$ we increase the first bin of each pallet to its c-fold. Most Executed (ME) and Least Leftover (LL) are at least as bad as MF, since we can produce instances leaving T_{opt} unchanged, where all pallets have the same number of bins by replacing the last bin of each pallet by as many bins as necessary.

Finally we consider the list $Q_3 = (q'''_1, q'''_2, q'''_3)$ of the following $k = 3$ sequences and $c = 5$, which also can be generalized.

$$q'''_1 = [a, b, c, d, e, f, \ldots], \quad q'''_2 = [b, c, d, e, f, \ldots], \quad q'''_3 = [c, d, e, f, a, \ldots]$$

An optimal pallet solution for the sequences in Q_3 is $T_{opt} = (a, b, c, d, e, f, \ldots)$ using $p = 2$ stack-up places. Early Closure (EC) starts to remove the first bin of q'''_1 by LR and then it removes bins from q'''_3 which leads to pallet solution $T_{EC} = (a, c, d, e, f, \ldots)$ using $p = c$ stack-up places.

Thus all of our strategies are not d-competitive for some constant d. Please note that the distribution of the bins of a pallet onto the sequences, which is a remarkable parameter in Sect. 4, can be enlarged by adding a bin of a new pallet at the end of an arbitrary number of sequences.

4 Experimental Study

Next we evaluate implementations of our on-line algorithms.

Creating Instances Since there are no benchmark data sets for the FIFO Stack-Up problem we generated random instances by an algorithm, which allows to give the following parameters: $p_{\max}$ the maximum number of stack-up places, m the number of pallets, k the number of sequences, $r_{\min}$ and $r_{\max}$ the minimum and maximum number of bins per pallet, and d the maximum number of sequences on which the bins of each pallet can be distributed. A detailed description of our algorithm for generating instances is given in [10].

Table 1 Performance (in percent) of our eight on-line algorithms for randomly generated instances of the FIFO Stack-Up problem. We achieved good results using the lookaheads of $c = 30$ for LL, $c = 10$ for MF, $c = 20$ for ME, and $c = 30$ for EC

Instance							Algorithm							
n	$p_{\max}$	m	k	$r_{\min}$	$r_{\max}$	d	LL	BF	MF	RD	ME	RR	LR	EC
1500	14	100	8	10	20	4	138	49	49	47	46	24	28	14
1500	14	100	8	10	20	6	84	54	54	60	33	20	21	9
1500	14	100	8	10	20	8	80	25	25	34	29	20	12	7
7500	18	300	10	15	35	4	257	107	107	73	38	42	38	21
7500	18	300	10	15	35	7	238	35	35	46	30	31	25	12
7500	18	300	10	15	35	10	146	27	27	57	28	27	22	10
17500	22	500	12	20	50	6	295	188	188	66	29	36	28	14
17500	22	500	12	20	50	9	122	22	22	48	27	30	21	10
17500	22	500	12	20	50	12	110	23	23	40	26	27	19	10
Average							163	59	59	52	32	29	24	12

Implementation and Evaluation We have implemented our on-line algorithms. In Table 1 we list some of our chosen parameters. For each assignment we randomly generated 100 instances, which are solved by our breadth first search solution from [4] to obtain an optimal number of stack-up places p_{opt} and by our eight on-line algorithms $A \in \{\text{RR}, \text{LR}, \text{BF}, \ldots\}$ to obtain a number of stack-up places p_A. In Table 1 the average performances $\frac{p_A - p_{\text{opt}}}{p_{\text{opt}}}$ are listed in percent.

Our evaluations show that Early Closure (EC) leads the best results using on average only 12% more stack-up places than an optimal off-line solution. We observe that for small distributions d the error of our on-line solutions is large. A reason for this is that for small values d the quotient $\frac{k}{d}$ is huge and thus the probability for opening a wrong pallet in a decision configuration becomes large.

References

1. de Koster, R.: Performance approximation of pick-to-belt orderpicking systems. Eur. J. Oper. Res. **92**, 558–573 (1994)
2. Rethmann, J., Wanke, E.: Storage controlled pile-up systems, theoretical foundations. Eur. J. Oper. Res. **103**(3), 515–530 (1997)
3. Gurski, F., Rethmann, J., Wanke, E.: Moving bins from conveyor belts onto pallets using FIFO queues. In: Operations Research Proceedings (OR 2013), Selected Papers, pp. 185–191. Springer (2014)
4. Gurski, F., Rethmann, J., Wanke, E.: A practical approach for the FIFO stack-up problem. In: Modelling, Computation and Optimization in Information Systems and Management Sciences, Advances in Intelligent Systems and Computing, vol. 360, pp. 141–152. Springer (2015)
5. Gurski, F., Rethmann, J., Wanke, E.: Algorithms for controlling palletizers. In: Operations Research Proceedings (OR 2014), Selected Papers, pp. 197–203. Springer (2016)

6. Gurski, F., Rethmann, J., Wanke, E.: On the complexity of the fifo stack-up problem. Math. Meth. Oper. Res. **83**(1), 33–52 (2016)
7. Gurski, F., Rethmann, J., Wanke, E.: An experimental study of algorithms for controlling palletizers. In: Operations Research Proceedings (OR 2015), Selected Papers, pp. 27–34. Springer (2017)
8. Borodin, A.: On-Line Computation and Competitive Analysis. Cambridge University Press (1998)
9. Fiat, A., Woeginger, G.: Online Algorithms: The State of the Art. LNCS, vol. 1442. Springer (1998)
10. Gurski, F., Rethmann, J., Wanke, E.: Integer programming models and parameterized algorithms for controlling palletizers. ACM Comput. Res. Repository (CoRR) (2015). arXiv:1509.07278

Solving an On-Line Capacitated Vehicle Routing Problem with Structured Time Windows

Philipp Hungerländer, Kerstin Maier, Jörg Pöcher, Andrea Rendl and Christian Truden

Abstract The capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW) is concerned with finding optimal tours for vehicles with given capacity constraints to deliver goods to customers within assigned time windows that can hold several customers and have a special structure (in our case equidistant and non-overlapping). In this work, we consider an on-line variant of the cVRPsTW that arises in the online shopping service of an international supermarket chain: customers choose a delivery time window for their order online, and the fleet's tours are updated accordingly in real time. This leads to two challenges. First, the new customers need to be inserted at a suitable place in one of the existing tours. Second, the new customers have to be inserted in real time due to very high request rates. This is why we apply a computationally cheap, two-step approach consisting of an insertion step and an improvement step. In this context, we present a Mixed-Integer Linear Program (MILP) and a heuristic that employs the MILP. In an experimental evaluation, we demonstrate the efficiency of our approaches on a variety of benchmark sets.

1 Introduction

The online market has been a growing sector for decades, and customers are increasingly interested in doing their weekly grocery shopping through the internet. This is why all main supermarket chains now provide online delivery services, where customers can select goods on the internet that are then delivered to their homes within a time window that the customer selects.

Online ordering poses new challenges to the grocery suppliers, since the customers select the delivery time window, and not the supplier. This makes organising the delivery fleet more difficult and leads suppliers to build their delivery schedule in an on-line fashion, where the tours/schedules of the delivery vans are updated as new customer orders come in. Moreover, all these steps have to be performed as

P. Hungerländer · K. Maier · J. Pöcher · A. Rendl · C. Truden (✉)
Department of Mathematics, Alpen-Adria Universität Klagenfurt, Klagenfurt, Austria
e-mail: christian.truden@aau.at

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_18

quickly as possible (within milliseconds) to provide a prompt online service to the customers.

In this paper, we tackle this problem in the context of a large international supermarket chain. The process of updating the vans' schedules is performed in recurring two steps, for each order a customer places online.

Insertion step The customer receives a selection of available time windows from which he can choose one for delivery. The larger the selection, the more satisfied the customer is with the service. The customer selects one of the available time windows and is accordingly inserted into the current delivery schedule.

Improvement step After the customer has successfully selected a time window, the system improves the current schedule by applying an improvement step. This step is essential to find as many feasible time windows as possible for the following customers and of course also to schedule as many customers as possible in total.

Within both above steps lies an optimization problem. The first one is concerned with inserting a customer optimally into an existing schedule, computing all time windows at which a given customer can be feasibly inserted. The second optimization problem is concerned with optimizing the existing, incomplete schedule, where the objective is to minimize the fleet's travel time without moving customers from their assigned time window. We denote this problem as the capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW).

For a recent, very good survey of the cVRPTW we refer to [1]. Toth and Vigo [5] give an overview over several types of the VRP including an extensive overview of different heuristics, integer programming approaches and case studies. Yang et al. [6] propose a closely related method to our simple insertion heuristic. Campbell and Savelsbergh [2] define the Home Delivery Problem (HDP), which is based on a similar application as our use case. However, they do not exploit the special time window structure and consider a different objective function. Finally, Ioannou et al. [3] define a similar real world problem, but for the traditional VRPTW.

In this short paper, we present a Mixed-Integer Linear Program (MILP) and a heuristic approach that deal both with the insertion and the improvement steps arising within our on-line variant of the cVRPsTW. Furthermore, we present preliminary experimental results on some benchmark instances.

This paper is organized as follows. In Sect. 2, we give a problem description, in Sect. 3 we outline approaches for inserting new customers into an existing schedule and present approaches for optimizing an incomplete schedule. In Sect. 4 we present computational results and conclude the paper.

2 Problem Description

The capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW) arises when delivering goods to customers who choose the delivery time window. The main difference to classical variants of the VRP are that the cus-

tomer chooses the time window for delivery, and that the time windows have a special structure (in our case equidistant and non-overlapping) that can be computationally exploited. We are given:

1. customers $a_i,\ i \in \mathcal{C}$, with assigned weights w and service times s,
2. a set of time windows $\mathcal{W}$, $|\mathcal{W}| = t$,
3. travel times for each pair of customers and
4. a finite set of tours $\mathcal{S} := \{\mathcal{A}, \mathcal{B}, \mathcal{C}, \ldots\}$, $|\mathcal{S}| = m$, with assigned capacities $C_i,\ i \in \mathcal{S}$.

The aim of the cVRPsTW is to find a feasible schedule with minimal travel time.

3 Inserting New Customers and Optimizing Tours

In our application, the customer places an order online, and the system proposes time windows during which the order can be delivered to the customer, who then selects a time window. In this section we outline our approach for finding this set of time windows, for which we use a simple insertion method, that takes a customer $\tilde{a}$, a tour $\mathcal{A}$ and a time window ω, and tries to insert $\tilde{a}$ into $\mathcal{A}$ at time window ω. Therefore, to calculate the set of available time windows for a new customer $\tilde{a}$, we apply the heuristic for each time window $\omega \in \mathcal{W}$ to each tour $\mathcal{A}$ in the schedule. Once the customer has selected a time window, we apply the simple insertion method again and choose from all feasible insertion points of the selected time window the one that leads to the smallest increase in travel time.

Facilitating calculations via earliest/latest arrival times. To facilitate the calculation of the feasible insertion points, we define the notion of an earliest and latest arrival time for each customer on a tour. It corresponds to the earliest, respectively the latest time, at which the van may arrive at a customer within time window ω, respecting the time windows of all other customers on the tour. When inserting customer $\tilde{a}$ between customers a_i and a_{i+1} we calculate the earliest and latest arrival time for $\tilde{a}$. These values are solely depended on the earliest, respectively the latest arrival time of the previous customer a_i and the following customer a_{i+1}, and time window ω.

Using the earliest and latest arrival time, a simple condition suffices to check if there is enough time between customer a_i and a_{i+1} to insert $\tilde{a}$ such that all customer orders can still be delivered within their assigned time windows. The condition is the following: customer $\tilde{a}$ can be inserted into a given tour between customer a_i and a_{i+1} in ω, if and only if the earliest arrival time of $\tilde{a}$ is less or equal than the latest arrival time of $\tilde{a}$. This condition allows to precompute the vast majority of the calculations that are needed to decide if the insertion results in a feasible tour.

Extending the heuristic with a MILP. The heuristic does not change the order of existing customers on the tour when inserting a new customer, therefore the insertion is not very advanced. In order to perform more sophisticated insertion operations, we utilize a MILP once the heuristic cannot find more feasible insertions.

We solve the Traveling Salesman Problems with structured Time Windows (TSPsTW) that is concerned with minimizing the travel times of a single tour $\mathcal{A}$ of the cVRPsTW. Hence in our setup, all tours in schedule $\mathcal{S}$, except $\mathcal{A}$, are fixed, and we solve the TSPsTW as feasibility problem (without objective function). In addition to the notation from Sect. 2, we use the following parameters and variables:

- a_1 is the start depot, a_n is the final depot and $\{a_2, \dots, a_{n-1}\}$ is the set of customers assigned to tour $\mathcal{A}$.
- $[n_i],\ i \in [t]$, are the customers assigned to time window i.
- $t_{ij},\ i \in [n_k], j \in [n_\ell], k, \ell \in [t], i \neq j, k \leq \ell \leq k+1$, is travel time from customer a_i to customer a_j plus service time at customer a_i.
- $b_i,\ i \in [t]$, and $e_i,\ i \in [t]$, are the start and end time of time window i respectively.
- $z_i \in \mathbb{R}^+,\ i \in [t]$, gives the wait time during time window i.
- $x_{ij} \in \{0,1\},\ i \in [n_k], j \in [n_\ell], k, \ell \in [t], i \neq j, k \leq \ell \leq k+1$, with the interpretation:

$$x_{ij} = \begin{cases} 1, & \text{if customer } j \text{ is visited right after customer } i, \\ 0, & \text{otherwise.} \end{cases}$$

The following constraints have to be satisfied after inserting a new customer into a given tour within a given time window:

$$\sum_{\substack{j \in [n_\ell], \ell \in [t] \\ j \neq i,\ k \leq \ell \leq k+1}} x_{ij} = 1, \qquad i \in [n_k],\ k \in [t],\ i \neq n, \tag{1}$$

$$\sum_{\substack{i \in [n_k],\ k \in [t] \\ i \neq j,\ k \leq \ell \leq k+1}} x_{ij} = 1, \qquad j \in [n_\ell],\ \ell \in [t],\ j \neq 1, \tag{2}$$

$$\sum_{\substack{i,j \in S, \\ i \neq j}} x_{ij} \geq |S| - 1, \quad \forall S \subset [n_k] \setminus \{s, f\},\ k \in [t],\ |S| \geq 2, \tag{3}$$

$$\sum_{\substack{i \in [n_k],\ j \in [n_\ell] \\ k \leq \ell \leq k+1,\ k < h,\ i \neq j}} t_{ij} x_{ij} + \sum_{i < h} z_i \geq b_h, \qquad h \in [t] \setminus \{1\}, \tag{4}$$

$$\sum_{\substack{i \in [n_k],\ j \in [n_\ell] \\ k \leq \ell \leq k+1,\ k, \ell \leq h,\ i \neq j}} t_{ij} x_{ij} + \sum_{i \leq h} z_i \leq e_h, \qquad h \in [t], \tag{5}$$

$$x_{ij} \in \{0,1\}, \qquad i \in [n_k],\ j \in [n_\ell],\ k, l \in [t],\ i \neq j,\ k \leq \ell \leq k+1, \tag{6}$$

$$z_i \geq 0, \qquad i \in [t]. \tag{7}$$

Equalities (1) and (2) ensure that all vertices except the final depot a_n have exactly one outgoing edge and all vertices except the start depot a_1 have exactly one ingoing edge. Inequalities (3) are the subtour elimination constraints that we do not add directly to our MILP but handle through separation. Finally inequalities (4) and (5)

guarantee that the arrival time of all customers is not before the start and not after the end of their assigned time window.

The most similar MILP formulation to our model is presented in [4], where, however, each time window contains only one customer. This is a critical difference to our version of the TSPTW, where several customers fit into a single time window, which we exploit in our approaches.

Optimizing tours. The improvement step follows the insertion step, and is applied on the tour $\mathcal{A}$ into which the new customer has been inserted. We add the following objective function to the MILP above in order to minimize the travel time and hence increase the chances to insert further customers into $\mathcal{A}$ at a later point:

$$\min \sum_{\substack{i\in[n_k],\ j\in[n_\ell],\ k,\ell\in[t] \\ k\le\ell\le k+1,\ i\neq j}} t_{ij}x_{ij} \quad . \tag{8}$$

4 Computational Experiments and Conclusion

In this section we present computational results. All experiments were performed on a Windows 7 64-bit machine equipped with an Intel Core i5-5300U (2×2300 MHz) and 12 GB RAM in single processor mode. We use Gurobi 6.5.1 as an IP-solver.

Benchmark instances. Our benchmark instances consist of customers where the coordinates on a square-grid are sampled from a two-dimensional uniform distribution and the travel times are calculated as the Euclidean distance between customers rounded to integers. Customer weights are sampled from a truncated normal distribution with mean of 7 and standard deviation of 2. Each customer is randomly assigned to one of the equidistant time windows. In our experiments we use instances, denoted e.g. as C100t7c150w5, consisting of 100/200/300 customers and 7 tours that have a capacity of 150/300/450 and 5/10/15 time windows each. Due to the size of the time windows and length of the service times of the customers there cannot be more than 6 customers within a time window per tour. The benchmark instances are designed to reflect real-world problems that arise in online shopping and can be downloaded from http://tinyurl.com/vrpstw.

Benchmark process. We iteratively insert new customers into the schedule, simulating customers placing orders online, where the preferred time window of the customer corresponds to the time window stated in the benchmark instance. Additionally we track how many time windows are available to each customer. We evaluate how many customers we are able to schedule as well as the runtime. Furthermore, we determine how many additional feasible time windows can be found when applying the MILP. For the improvement step we measure the improvement of the schedule compared to the schedule directly after each insertion step.

Our results summarized in Table 1 show that all our approaches use very little time (as required) and provide satisfactory results with respect to solution quality: we are able to determine insertion points for most time windows via our simple insertion heuristic and the MILP yields further improvements in both steps.

Table 1 We state average values over five benchmark instances each. On the one hand we compare insertion via our simple insertion heuristic and via our MILP feasibility problem. On the other hand we present the average improvement of the objective function per iteration obtained by our MILP optimization problem

Approach	Per step	C100t7c150w5	C200t7c300w10	C300t7c450w15
Heuristic-insertion	Av. no. of windows	4.884	9.858	14.835
	Av. time	0.17 ms	0.23 ms	0.23 ms
MILP-insertion	Av. no. of add. windows	0.016	0.023	0.046
	Av. time	8.23 ms	23.64 ms	49.98 ms
MILP-optimization	Av. impr. over insert. step	0.158%	0.096%	0.061%
	Av. time	7.49 ms	13.56 ms	21.51 ms

Conclusion. In this paper we presented the capacitated Vehicle Routing Problem with structured Time Windows (cVRPsTW) that arises in the context of delivering goods, where customers choose the delivery time window, and the delivery schedule is updated as new customers arrive. We introduced a two-step approach where we employ a heuristic and a Mixed-Integer Linear Program (MILP) in the respective insertion and improvement step. Our computational evaluation demonstrates that our approaches comply with the strict time limits and can produce good results within milliseconds, rendering them applicable to a real-world setting. For future research it would be interesting to extend our approach to further, more advanced heuristics and a MILP that improves the delivery schedule for a specific time window over all tours.

References

1. Baldacci, R., Mingozzi, A., Roberti, R.: Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. Eur. J. Oper. Res. **218**(1), 1–6 (2012)
2. Campbell, A.M., Savelsbergh, M.W.P.: Decision support for consumer direct grocery initiatives. Transp. Sci. **39**(3), 313–327 (2005)
3. Ioannou, G., Kritikos, G., Prastacos, G.: A greedy look-ahead heuristic for the vehicle routing problem with time windows. J. Oper. Res. Soc. **52**(5), 523–537 (2001)
4. Lodi, A., Milano, M.: A hybrid exact algorithm for the tsptw. INFORMS J. Comput. **14**, 403–417 (2002)
5. Toth, P., Vigo, D. (eds.): The Vehicle Routing Problem. Society for Industrial and Applied Mathematics (2002)
6. Yang, X., Strauss, A.K., Currie, C.S.M., Eglese, R.: Choice-based demand management and vehicle routing in e-fulfillment. Transp. Sci. (2014)

On the Solution of Generalized Spectrum Allocation Problems

John Martinovic, Eduard Jorswieck and Guntram Scheithauer

Abstract We consider a spectrum aggregation based spectrum allocation problem (SAP) for coexisting wireless systems: find the maximum number of secondary users whose bandwidth requirements can be satisfied by aggregating (parts of) given spectrum holes. In the classical form, this optimization problem turns out to share a common structure with the one-dimensional skiving stock problem (SSP), where as many (large) items as possible have to be constructed simultaneously by combining (smaller) items of a given supply. However, in practice, the spectrum aggregation is usually restricted by hardware limitations, such as filter technologies, and the capability of controlling interference. These additional constraints separate the considered problem from an ordinary SSP, and represent a new challenge in the field of discrete optimization. This article provides a general introduction to the relations between the SSP and the SAP. Moreover, we will discuss, how practically meaningful extensions of the classical SAP can be tackled from a mathematical point of view. As a main contribution, we exploit some important problem-specific properties to derive tailored solution techniques.

This work is supported in part by the German Research Foundation (DFG) within the Collaborative Research Center SFB 912 HAEC.

J. Martinovic (✉) · G. Scheithauer
Institute of Numerical Mathematics, SFB 912 – HAEC,
Technische Universität Dresden, 01062 Dresden, Germany
e-mail: john.martinovic@tu-dresden.de

G. Scheithauer
e-mail: guntram.scheithauer@tu-dresden.de

E. Jorswieck
Chair of Communications Theory, SFB 912 – HAEC,
Technische Universität Dresden, 01062 Dresden, Germany
e-mail: eduard.jorswieck@tu-dresden.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_19

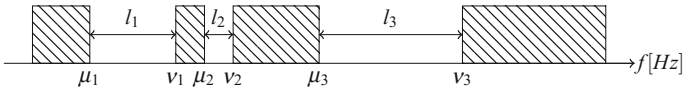


Fig. 1 A possible frequency band with spectrum holes of bandwidths $l_i = \nu_i - \mu_i$ $(i = 1, 2, 3)$

1 Introduction

Due to the significantly increased importance of wireless connectivity in the last years, the natural radio spectrum has become a very important and scarce resource. Normally, it is regulated by governmental entities and fixed parts of it are assigned to licensed holders for a long time, see [3]. However, for large parts of the (licensed) spectrum, the utilization is quite low leading to many wasted vacant frequency intervals [2], see Fig. 1 for a schematic.

Typically, the *spectrum holes* are too small to meet the bandwidth requirements of *secondary users* (SUs). However, Software Defined Radio (SDR) provides a flexible and programmable transceiver structure that allows to combine vacant intervals in order to obtain sufficiently large transmission channels [4]. In this regard, the SAP asks for the maximal number of SUs that can be assigned to the given spectrum holes such that the bandwidth requirement of each SU is satisfied.[1] Note that the spectrum aggregation itself is usually restricted by hardware limitations, such as filter technologies, and the capability of controlling interference, separating the considered problem from an ordinary SSP [7, 10].

In the next section, we aim at providing a short introduction to the SAP and its relation to the SSP, respectively. Thereafter, we focus on the incorporation of practically meaningful constraints in order to obtain application-oriented extensions of the standard problem described in [6]. As a main contribution, we show how problem-specific properties can be used within a solution strategy. Note that, due to the space limitation, this paper only deals with one selected topic of the submitted presentation. The proofs of the results and further contributions can be found online [5].

2 Preliminaries

Consider a frequency band where some portions are already covered by licensed users, see Fig. 1. The unoccupied spectrum holes shall be used by SUs each of which having a required bandwidth of $R \in \mathbb{N}$ (typically in kHz or MHz). Thereby, the SUs may aggregate (parts of) the given spectrum holes in order to obtain sufficiently large transmission channels. Assuming each spectrum hole to be accessible by at most one SU (which might be appropriate to suppress interference in some particular

[1] Similar questions do also arise when saving data on hard drive disks or when managing inventory in storehouses.

applications), the above stated problem refers one-to-one to an ordinary SSP and off-the-shelf models [7] are directly applicable. But, due to hardware limitations such as filter technologies in the radio frequency (RF) chains, only spectrum holes that are in a certain neighborhood to each other, specified by a *maximal aggregation range (MAR)* $\delta \in \mathbb{N}$, can be combined. In that case, modifications of the known approaches can be used, leading to allocations making use of up to about 90% of the total vacant bandwidth, see [6].

In this article, the problem of several SUs per spectrum hole, hereinafter referred to as the *generalized spectrum allocation problem* (GSAP), is considered. To this end, we may rephrase the definition of an instance as follows.

Definition 1 A tuple $E = (n, \mu, \nu, R, \delta)$ with $n \in \mathbb{N}$ spectrum holes, $\mu, \nu \in \mathbb{Z}_+^n$ representing the initial and terminating frequencies, $R, \delta \in \mathbb{N}$ denoting required bandwidth and the MAR, respectively, is called *instance (of the GSAP)* if and only if $\mu_1 < \nu_1 < \ldots < \mu_n < \nu_n$ holds.

Let $I := \{1, \ldots, n\}$. Then a spectrum hole $[\mu_i, \nu_i]$ ($i \in I$) can be shared between several SUs as long as they operate in pairwise distinct subsets of $[\mu_i, \nu_i]$.

Definition 2 A triple $(a, p, q) \in \mathbb{B}^n \times \mathbb{Z}_+^n \times \mathbb{Z}_+^n$ is called *(allocation) pattern*, if $e^\top(q - p) = R$ (with $e = (1, \ldots, 1)^\top \in \mathbb{Z}_+^n$) holds, and if the following conditions are satisfied:

(a) $\mu \le p, p + a \le q$, and $q_i \le a_i \nu_i + (1 - a_i)\mu_i$ for all $i \in I$,
(b) $\overline{q} - \underline{p} \le \delta$ where $\overline{q} := \max\{q_i \mid q_i > p_i\}$ and $\underline{p} := \min\{p_i \mid q_i > p_i\}$.

Here $[p_i, q_i)$ describes the specific subset of $[\mu_i, \nu_i]$ that is occupied by the pattern (a, p, q). Due to (a) we obtain $[p_i, q_i) = [\mu_i, \mu_i) = \emptyset$ if and only if $a_i = 0$. Otherwise (for $a_i = 1$), these conditions ensure $\mu_i \le p_i < q_i \le \nu_i$. Moreover, $e^\top(q - p) = R$ satisfies the bandwidth condition, whereas condition (b) specifies the δ-closeness. Obviously, the latter depends on two further optimization problems, and is, hence, rather inappropriate for a good description of the pattern set.

3 A Solution Strategy Based on Connected Patterns

As we have seen, the formulation of a pattern-based ILP for the GSAP is rather difficult due to the fact that the pattern set itself does likely not possess a practical (at best linear) description that could efficiently be used in a column generation algorithm. Instead, we focus on some theoretical observations in order to find a solution technique that exploits the problem-specific properties.

Definition 3 Let (a, p, q) be a pattern. A point $\sigma \in \bigcup_{i=1}^n [\mu_i, \nu_i)$ is called *interruption point* of (a, p, q) if $\underline{p} < \sigma < \overline{q}$ and $[\sigma, \sigma + 1) \cap [p_i, q_i) = \emptyset$ for all $i \in I$ hold. A pattern (a, p, q) that does not contain any interruption point is called *connected*.

Based on this, [5, Fig. 3] motivates an important principle of our theoretical study.

Lemma 1 *Let (a, p, q) be a pattern. Then there exists a unique connected pattern $(\widetilde{a}, \widetilde{p}, \widetilde{q})$ with $\underline{p} = \underline{\widetilde{p}}$.*

According to this lemma, we can conclude:

Theorem 1 *There exists a solution of the GSAP that only consists of connected patterns.*

Thus, considering connected patterns is sufficient to solve the GSAP. As a next step, note that for each connected pattern (a, p, q) with $\underline{p} \notin \{\mu_1, \dots, \mu_n\}$ we can shift the whole pattern one unit to the left (by respecting the given spectrum holes) obtaining a new feasible pattern (a', p', q') with $\underline{p'} = \underline{p} - 1$, see [5, Fig. 6].

Definition 4 The pattern (a', p', q') is called *left-shift* of (a, p, q).

After a finite number of left-shifts, we obtain a pattern (a', p', q') with $\underline{p'} = \mu_i$ for some $i \in I$, leading to the observation:

Theorem 2 *There exists a solution of the GSAP, containing only connected patterns, where each pattern either starts at some point of $\{\mu_1, \dots, \mu_n\}$ or at the endpoint $\bar{q}$ of the previous pattern.*

Hence, a solution of this problem can be obtained by a successive assignment of connected patterns to the given spectrum holes while respecting the δ-closeness. To ease the notation, let $U(x, y) = \sum_{i=1}^{n} \left|[x, y] \cap [\mu_i, \nu_i)\right|$ with $|[a, b)| = \max\{b - a, 0\}$ denote the total vacant bandwidth between the points $x \in \mathbb{N}$ and $y \in \mathbb{N}$. Then, the following algorithm leads to a solution of the GSAP:

Algorithm 1 Solving the GSAP

Input: Instance $E = (n, \mu, \nu, R, \delta)$.
1: Set $k := 1, j := 1, x^j := \mu_1, \mu_{n+1} := \nu_n$ and $z := 0$.
2: **while** $k \leq n$ and $U(x^j, \nu_n) \geq R$ **do**
3: **if** $U(x^j, x^j + \delta) \geq R$ **then**
4: Compute $p \in \{k, \dots, n\}$ and $y^j \in (\mu_p, \nu_p]$ with $U(x^j, y^j) = R$. Save (x^j, y^j).
5: Set $j := j + 1, z := z + 1, x^j := y^{j-1}, k := p$ (or $x^j := \mu_{p+1}, k := p + 1$ if $y^{j-1} = \nu_p$).
6: **else**
7: Set $k := k + 1, x^j := \mu_k$.
8: **end if**
9: **end while**
Output: optimal value z, saved pairs (x^j, y^j) for $j = 1, \dots, z$.

4 The Role of Interference and Heterogeneous Bandwidths

Algorithm 1 provides a solution where (possibly) two patterns (two SUs) are placed directly next to each other in some given spectrum hole. Due to the fact that a signal in some subset $[p_i, q_i)$ of $[\mu_i, \nu_i]$ is not ideal in practice, it needs a certain additional

range $[p_i - \varepsilon, p)$ and $[q_i, q_i + \varepsilon)$ (for some small $\varepsilon \in \mathbb{N}$) outside of $[p_i, q_i)$ to decay completely, see [5, Fig. 7]. Hence, if two patterns are positioned directly one after the other, their signals will interfere in a small neighborhood of their joint borderline. Fortunately, this problem can be tackled by leaving a certain spectral distance (for instance an interval of width ε) between two neighboring patterns, if and only if they operate in the same spectrum hole. Such unoccupied frequency ranges are called *guard bands*, see [5, Fig. 8] for a schematic and [1] or [9] for a more detailed explanation. Note that this interference does also occur between SUs and licensed holders. Here, the size of the spectrum holes can be changed to $[\widetilde{\mu}_i, \widetilde{\nu}_i]$ with $\widetilde{\mu}_i := \mu_i + \varepsilon$ and $\widetilde{\nu}_i := \nu_i - \varepsilon$ prior to the optimization.

Theorem 3 *There exists a solution of the GSAP, containing only connected patterns, with guard bands, where each pattern either starts at some point of* $\{\widetilde{\mu}_i \mid i \in I\}$ *or with a distance of* ε *to the endpoint of the previous pattern.*

Consequently, Algorithm 1 can also be applied after some minor modifications:

Algorithm 2 Solving the GSAP with guard bands

Input: Instance $E = (n, \mu, \nu, R, \delta)$, guard band width $\varepsilon \in \mathbb{N}$.
Set $k := 1, j := 1, x^j := \mu_1$, and $z := 0$.
Compute $\widetilde{\mu}_i, \widetilde{\nu}_i$ for all $i \in I$, and define $\widetilde{\mu}_{n+1} := \widetilde{\nu}_n$.
while $k \leq n$ and $U(x^j, \widetilde{\nu}_n) \geq R$ **do**
    if $U(x^j, x^j + \delta) \geq R$ **then**
        Compute $p \in \{k, \ldots, n\}$ and $y^j \in (\widetilde{\mu}_p, \widetilde{\nu}_p]$ with $U(x^j, y^j) = R$.
        Save (x^j, y^j) and set $j := j + 1, z := z + 1$.
        if $\widetilde{\nu}_p - y^{j-1} \leq \varepsilon$ **then**
            Set $k := p + 1, x^j = \widetilde{\mu}_k$.
        else
            Set $x^j = y^{j-1} + \varepsilon$ and k:=p.
        end if
    else
        Set $k := k + 1, x^j := \widetilde{\mu}_k$.
    end if
end while
Output: optimal value z, saved pairs (x^j, y^j) for $j = 1, \ldots, z$.

So far we assumed all SUs to possess the same required bandwidth R. Due to standardization in wireless communications this situation is not completely implausible in practice. Nevertheless, this assumption may represent a too strong restriction when focusing on an application-oriented framework. For a given number $N \in \mathbb{N}$ of SUs, we may assume that $R_1 \leq R_2 \ldots \leq R_N$ holds for their specific required bandwidths. Then the following result can be obtained:

Theorem 4 *Let* z *represent the optimal value of the GSAP with specific bandwidths. Then there exists a solution where the requirements* $R_1, \ldots, R_z$ *are satisfied.*

However, note that finding the concrete positionings of the chosen secondary users is much more difficult than in the previous cases.

5 Conclusions and Outlook

In this article, we considered the SAP as a generalized version of the SSP. As a main contribution, we have seen how different practically meaningful constraints can be tackled in the spectrum allocation framework. Although it is much harder to find an appropriate modeling approach (for instance a pattern-based ILP), the problems become somehow more managable, i.e., the problem-specific properties can easily be exploited to find a possible solution by means of the presented algorithms.

One main challenge for our future work consists in the consideration of time-dependent spectrum holes and other objective functions [8], such as maximizing the energy efficiency of the obtained allocation. This goal represents a main research area of the Collaborative Research Center Highly Adaptive Energy-efficient Computing (HAEC),[2] therefore being, in general, of high relevance in our current and future research.

References

1. Bany Salameh, H.A., Krunz, M., Manzi, D.: Spectrum bonding and aggregation with guard-band-awareness in cognitive radio networks. IEEE Transact. Mob. Comput. **13**(5), 569–581 (2014)
2. Federal Communications Commission. Spectrum policy task force: report. ET-Docket 02-135. (2002) http://www.fcc.gov/sptf
3. Haykin, S.: Cognitive radio: brain-empowered wireless communications. IEEE J. Sel. Areas Commun. **23**(2), 201–220 (2013)
4. Lee, H., Vahid, S., Moessner, K.: A survey of radio resource management for spectrum aggregation in LTE-advanced. IEEE Commun. Surv. Tutorials **16**(2), 745–760 (2014)
5. Martinovic, J., Jorswieck, E., Scheithauer, G.: The skiving stock problem and its application to resource allocation. Preprint MATH-NM-01-2016. Technische Universität Dresden (2016)
6. Martinovic, J., Jorswieck, E., Scheithauer, G.: The skiving stock problem and its application to cognitive radio networks. IFAC Papers Online, Proceedings of the 8th IFAC Conference on Manufacturing Modelling, Management and Control **49**(12), 99–104 (2016)
7. Martinovic, J., Scheithauer, G.: Integer linear programming models for the skiving stock problem. Eur. J. Oper. Res. **251**(2), 356–368 (2016)
8. Tragos, E.Z., Zeadally, S., Fragkiadakis, A.G., Siris, V.A.: Spectrum assignment in cognitive radio networks: a comprehensive survey. IEEE Commun. Surv. Tutorials **15**(3), 1108–1135 (2013)
9. Uyanik, G.S., Abdel-Rahman, M.J., Krunz, M.: Optimal channel assignment with aggregation in multi-channel systems: a resilient approach to adjacent-channel interference. Ad Hoc Netw. **20**, 64–76 (2014)
10. Zak, E.J.: The skiving stock problem as a counterpart of the cutting stock problem. Intern. Trans. Oper. Res. **10**, 637–650 (2003)

[2] https://tu-dresden.de/sfb912.

Two-Stage Cutting Stock Problem with Due Dates

Zeynep Sezer and İbrahim Muter

Abstract In this study, we consider a scheduling extension for the two-stage cutting stock problem with the integration of order due dates. The two-stage cutting stock problem arises when technical restrictions inhibit demanded items to be cut from stock rolls directly, and hence require the cutting process to be done in two subsequent stages. The mathematical model proposed for the due date extension aims to determine a cutting plan which not only minimizes the number of stock rolls used but also reduces tardiness and earliness costs incurred. Preliminary results have shown that the modeling approach used is capable of overcoming difficulties caused by the dependencies between stages.

1 Introduction

The one-dimensional multi-stage cutting stock (MSCS) problem arises when demanded items of different widths are required to be cut from stock rolls in multiple stages due to technical restrictions. In these problems, rolls produced at each stage are used as an input for the subsequent stage. The MSCS problems studied in the literature are generally inspired by real problems encountered in the paper and the film industries. In this paper, we focus on the two-stage version of this problem which will be referred to as the MSCS problem from hereafter.

The compact model for the MSCS problem was first introduced in [7], which consisted of two types of cutting patterns, one for each stage of the cutting process. The difficulty in solving this model stems from the unknown widths of rolls produced in the intermediate stage, referred to as the intermediate rolls, which correspond to a set of linking constraints. Hence, the application of traditional column

Z. Sezer (✉)
Department of Industrial Engineering, Bahçeşehir University, Beşiktaş, 34353 Istanbul, Turkey
e-mail: zeynep.sezer@bahcesehir.edu.tr

İ. Muter
School of Management, University of Bath, Claverton Down, Bath BA2 7AY, UK
e-mail: i.muter@bath.ac.uk

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_20

generation falls short since the generation of a new pattern consisting of currently missing intermediate rolls introduces linking constraints whose corresponding dual variables are unknown. To solve the linear programming (LP) relaxation of the MSCS problem, [7] developed a heuristic column-and-row generating algorithm which generates one intermediate roll at each iteration. Later, [3] characterized the LP model as a column-dependent-rows problem, which is a generic class of problems identified by a set of linking constraints that are dependent on a set of variables. In their study, they proposed a generic simultaneous column-and-row generation algorithm which correctly prices out columns in the absence of some linking constraints using a novel row-generating pricing subproblem.

The cutting stock problems are generally concerned with the optimization of cutting processes so that the trim loss incurred during these operations are minimized. However, in practice, customers place orders with specific due dates, and on such circumstance, the costs incurred due to a poor scheduling decision, such as costs associated with late orders or lost sales, may surpass the costs of raw material wasted during a cutting process. Therefore, we aim to develop an integrated mathematical model which allows for making coordinated decisions on both cutting and scheduling. In the literature, various scheduling objectives are considered for the single-stage cutting stock problem which are handled both subsequent to the generation of patterns ([2, 5, 6]) and simultaneously with integrated models ([1, 4]). No scheduling extension to the MSCS problem has yet been investigated. We consider an extended MSCS problem, in which order due dates and associated costs are incorporated to the compact formulation. These costs include both tardiness and earliness costs, the latter being vital for minimizing the work-in-process inventories of intermediate rolls.

2 Problem Statement

The due date extended MSCS problem can be described as follows: Within a production planning horizon indexed by $q \in Q$, a set of customer orders, each of which consists of one type of finished roll, $j \in J$, are to be satisfied by their due dates through a two-stage cutting process. In the first stage, identical stock rolls are cut into intermediate rolls, indexed by $i \in I$, through first stage cutting patterns, indexed by $k \in K$. The widths of intermediate rolls are not known in advance but need to lie within an interval, $[s^{min}, s^{max}]$. In the second stage, the intermediate rolls produced in the first stage are cut into finished rolls to satisfy the demand on finished rolls through second stage cutting patterns, indexed by $n \in N$. The demand on an order not satisfied by its corresponding due date is considered tardy and is penalized in the objective function. It is assumed that the stock rolls and the intermediate rolls are cut on separate consecutive machines, forming a flow-shop scheduling environment. Thus, there is a precedence relationship between the two stages in such a way that the intermediate rolls to be cut in the second stage within a production period must be made available until the end of that period through first-stage cutting patterns.

The main difficulty in solving integrated cutting and scheduling problems is caused by the conflicting nature of their objectives, rendering a multi-objective optimization problem. Furthermore, the time dependent scheduling components are inconsistent with those of the cutting problem which are usually defined in terms of the number of pieces or patterns cut. Unlike the scheduling problems, the completion time of an order is dependent on the type and the number of the patterns cut in the production periods. To overcome these difficulties, [4] introduced an alternative modeling approach for the due date extension of the single-stage cutting stock problem. In their approach, they discretized the planning horizon into irregular production periods using order due dates, $q \in Q$. The time intervals are expressed in terms of the machine capacity, which is the number of stock rolls a cutting machine can process within that period. Their model identifies tardiness as the number of stock rolls cut over the machine capacity to complete an order, which is penalized in the objective function.

On the other hand, as later argued by [1], the tardiness costs returned by the formulation in [4] are not always exact since the demand on each finished roll is required to be satisfied through the cutting patterns cut until the end of the associated production period. Consequently, a tardy order in a production period causes orders in subsequent periods to be delayed which results in tardiness costs to be overcalculated. They suggested an alternative formulation by combining it with a lot-sizing problem, which allowed demand to be satisfied through rolls produced in any production period. In their model, negative inventory levels indicate items being backlogged and are used to identify orders that are late. The exact tardiness values are obtained through a refinement procedure by partitioning the production periods.

3 Mathematical Model

In the extended MSCS problem, the cutting process of stock rolls and the intermediate rolls are completed on two separate machines consecutively whose capacities at each period q, Δ_q^1 and Δ_q^2, are defined in terms of the number of stock rolls and the number of intermediate rolls that can be cut during that period, respectively. We define the number of stock rolls cut at each period q with y_{qk}, $k \in K$ and the number of intermediate rolls cut at each period q with x_{qn}, $n \in N$. When $\Delta_q^1 \approx \Delta_q^2$, i.e. the processing time of a stock roll and an intermediate roll are similar, the machine in the second stage becomes the bottleneck of the complete process: since a first stage cutting pattern typically contains several intermediate rolls, the number of rolls that need to be processed by the second machine is higher. In such cases, intermediate rolls required in the second stage are produced earlier than necessary and stocked to be used at a later period which results in work-in-process inventory. In order to alleviate this problem, we define variable $sf_{q,i} \in \mathbb{R}_+$ for the inventory level of intermediate roll $i \in I$ at the end of period $q \in Q$, which is penalized in the objective function to ensure that these rolls are cut through first-stage cutting patterns

in the respective periods when demanded by the second stage. These variables are not allowed to take negative values to conform to the precedence relation between the first and the second-stage cutting operations. Even though there is no actual due date on the intermediate rolls, the precedence relation between stages constitutes a self-imposed due date, which causes earlier production of intermediate rolls to be penalized. The inventory level for finished roll $j \in J$ at the end of period $q \in Q$ is denoted by $ss_{q,j} \in \mathbb{R}$. The positive and negative values of $ss_{q,j}$ correspond to items of finished roll j being stored and backlogged, respectively, at the end of period q. The demand on finished roll j at period q is denoted by parameter d_{qj} with $d_{qj} = d_j$, if $q = j$, and $d_{qj} = 0$, otherwise. If finished roll j is late, meaning that $s_{q,j} < 0$ in some time period $q \geq j$, binary variable t_{qj} takes a value of one, which inflicts a tardiness cost in the objective function. Therefore, the due date extension of the MSCS problem has three objectives, namely the minimization of the number of stock rolls (trim loss), the tardiness cost and the inventory cost of the intermediate rolls (earliness cost). In the mathematical model given below, we use a prominent method of multi-objective optimization, named weighted-sum method, to reach a single objective problem in which the objectives are multiplied with α, β, and γ.

$$\text{Minimize} \quad \alpha \sum_{q\in Q}\sum_{k\in K} y_{qk} + \beta \sum_{j\in J}\sum_{q>j} \Delta_q^2 t_{qj} + \gamma \sum_{i\in I}\sum_{q\in Q} sf_{q,i}, \tag{1}$$

$$\text{subject to} \quad \sum_{k\in K} y_{qk} \leq \Delta_q^1, \quad q \in Q, \tag{2}$$

$$\sum_{n\in N} x_{qn} \leq \Delta_q^2, \quad q \in Q, \tag{3}$$

$$ss_{q-1,j} + \sum_{n\in N} B_{jn} x_{qn} = ss_{q,j} + d_{qj}, \quad j \in J, q \in Q, \tag{4}$$

$$sf_{q-1,i} + \sum_{k\in K} C_{ik} y_{qk} = sf_{q,i} + \sum_{n\in N} D_{in} x_{qn}, \quad i \in I, q \in Q, \tag{5}$$

$$d_j t_{qj} - ss_{q-1,j} \geq 0, \quad j \in J, q > j, \tag{6}$$

$$y_{qk}, x_{qn}, sf_{qi} \geq 0, \ integer, \quad n \in N, k \in K, q \in Q, \tag{7}$$

$$t_{qj} \in \{0, 1\}, \quad j \in J, q \in Q, \tag{8}$$

$$ss_{qj} \ urs, \ integer, \quad j \in J, q \in Q, \tag{9}$$

where C_{ik} shows the number of intermediate roll $i \in I$ existing in the first-stage cutting pattern $k \in K$, and B_{jn} denotes the number of finished roll $j \in J$ existing in the second-stage cutting pattern $n \in N$. Moreover, $D_{in} = -1$, only if the second-stage cutting pattern $n \in N$ is cut from intermediate roll i. Constraints (2) and (3) restrict the number of the first- and the second-stage cutting patterns cut in a period with the available capacity of the machine in the first and the second stages, respectively. The equilibrium constraints (4) guarantee that for each finished roll $j \in J$, the demand and the items carried to the next period are satisfied with the second-stage cutting patterns produced in that period plus the available inventory remaining from the previous period. Similarly, constraints (5) which link the two stages ensure that the

number of intermediate roll i produced at period q and carried from the previous period is equal to the number of second-stage cutting patterns cut from i plus the work-in-process inventory of this roll carried to the next period. Constraint set (6) links binary variable t_{qj} with $ss_{q-1,j}$ in such a way that the negative values of the latter impose the former to one.

4 Computational Experiments

In this section, we perform tests to evaluate the results obtained from our proposed model. The number of variables and constraints in (1)–(9) are dependent on the number of intermediate rolls and the number of periods. The structure of this problem is amenable to simultaneous column-and-row generation which is out of the scope of this paper. Hence, we employ a random instance generator that allows pre-enumeration of all cutting patterns, which also reveals the complete set of intermediate rolls. To that end, we choose $|J| = 5$ whose widths and demands are randomly generated from U(300, 800) and U(10, 110), respectively. Stock roll width and the limits on the intermediate rolls are $W = 5000$, $s^{max} = 1700$ and $s^{min} = 1400$, respectively. Second stage machine capacity for each period Δ_q^2 is determined randomly between the minimum and the maximum number of intermediate rolls required to complete an order, U($\left\lfloor \sum_j \alpha_j d_j / s^{max} \right\rfloor$, $\left\lceil \sum_j \alpha_j d_j / s^{min} \right\rceil$). First stage machine capacities at each period, Δ_q^1, are scaled according to Δ_q^2, in particular $\Delta_q^1 = \Delta_q^2$ and $3\Delta_q^1 = \Delta_q^2$. The parameters in the objective function are selected as $\alpha = 1$, $\beta = 1$ and $\gamma = \{0, 0.005\}$. While the weights on the trim loss and the tardiness objectives are equal, the tardiness costs at each period, which is multiplied by the second stage machine capacities, are emphasized more in the objective function. This parameter selection is a result of our assumption that the lexicographic order of importance of the objectives is tardiness, trim loss and intermediate roll inventory. The experiments are conducted on a computer with a 1.60 GHz Intel Core i5-4200U Processor and 4 GB of RAM. The algorithms are implemented on C++ using the MIP solver of CPLEX 12.5 and Concert 2.5.

The numerical results for ten randomly generated instances are reported in Table 1, stating respectively, the objective function value {the number of stock rolls used, total tardiness, total inventory of intermediate rolls}. These results show that when the capacities of the two machines are similar, the stock rolls are cut in advance of the periods for which the produced intermediate rolls are needed. This can be observed by the higher work-in-process inventory levels in columns 1 and 3 compared to columns 2 and 4, respectively. In addition, decreasing the capacity of the first stage machine causes fewer number of stock rolls to be cut at each period which results in increased tardiness of orders and reduced work-in-process inventory. Penalizing the early production of intermediate rolls in the objective function, even with a very small coefficient of $\gamma = 0.005$, has a considerable effect on the work-in-inventory levels which is best observed from the results in columns 1 and 3.

Table 1 Experimental results

	$\gamma = 0$		$\gamma = 0.005$	
	$\Delta_q^1 = \Delta_q^2$	$3\Delta_q^1 = \Delta_q^2$	$\Delta_q^1 = \Delta_q^2$	$3\Delta_q^1 = \Delta_q^2$
1	124{46, 78, 153}	145{46, 99, 9}	124, 02{46, 78, 4}	145{46, 99, 0}
2	28{28, 0, 54}	65{28, 37, 0}	28, 015{28, 0, 3}	65{28, 37, 0}
3	111{38, 73, 178}	111{38, 73, 5}	111{38, 73, 0}	111{38, 73, 0}
4	104{46, 58, 65}	142{46, 96, 2}	104{46, 58, 0}	142{46, 96, 0}
5	142{43, 99, 87}	166{43, 123, 1}	142, 015{43, 99, 3}	166{43, 123, 0}
6	109{39, 70, 155}	109{39, 70, 0}	109{39, 70, 0}	109{39, 70, 0}
7	78{35, 43, 177}	78{35, 43, 4}	78{35, 43, 0}	78{35, 43, 0}
8	110{35, 75, 123}	174{36, 138, 31}	110, 335{35, 75, 67}	174{36, 138, 0}
9	122{27, 95, 130}	133{27, 106, 1}	122, 005{27, 95, 1}	133{27, 106, 0}
10	127{35, 92, 70}	143{35, 108, 2}	127, 015{35, 92, 3}	143{35, 108, 0}

5 Conclusion and Future Work

In this paper, we present a modeling approach to integrate order due dates in the MSCS problem. Our mathematical model incorporates three objectives into a single objective function, and is capable of reducing the work-in-process inventory considerably, while at each period, selecting the cutting patterns for the two stages that minimize the tardiness and the trim loss in this order. As a future research, we intend to develop a simultaneous column-and-row generation algorithm to solve larger instances of this problem. Considering the difficulty of this problem, we also strive to develop a heuristic approach to obtain good feasible solutions.

Acknowledgements This work has been partially completed while the author was a member in the Faculty of Engineering at Bahçeşehir University.
This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 113M480.

References

1. Arbib, C., Marinelli, F.: On cutting stock with due dates. Omega-Int. J. Manage. S. **46**, 11–20 (2014)
2. Dyson, R.G., Gregory, A.S.: The cutting stock problem in the flat glass industry. Oper. Res. Quart. **25**(1), 41–53 (1974)
3. Muter, I., Birbil, I., Bülbül, K.: Simultaneous column-and-row generation for large-scale linear programs with column-dependent-rows. Math. Program. **142**(1–2), 47–82 (2013)
4. Reinertsen, H., Vossen, T.W.M.: The one-dimensional cutting stock problem with due dates. Eur. J. Oper. Res. **201**(3), 701–711 (2010)
5. Yanasse, H.H.: On a pattern sequencing problem to minimize the maximum number of open stacks. Eur. J. Oper. Res. **100**(3), 454–463 (1997)

6. Yuen, B.J.: Improved heuristics for sequencing cutting patterns. Eur. J. Oper. Res. **87**(1), 57–64 (1995)
7. Zak, E.J.: Row and column generation technique for a multistage cutting stock problem. Comput. Oper. Res. **29**(9), 1143–1156 (2002)

Part VI
Energy and Environment

A Two-Stage Heuristic Procedure for Solving the Long-Term Unit Commitment Problem with Pumped Storages and Its Application to the German Electricity Market

Alexander Franz and Jürgen Zimmermann

Abstract In electricity systems unit commitment problems (UCP) target at a proper scheduling and coordinating of thermal plants, renewable energies, and storages. The need for fast solution methods has been growing in line with recent changes in the electricity system's environment and complexity, in particular with the increasing share of volatile renewable feed-ins. In order to meet this need even for large-scale systems a decomposition methodology for the UCP is suggested within this paper. Our two-stage decomposition first performs an isolated dispatching of thermal plants using a greedy algorithm, rule-based algorithms and local search based steps, followed by a re-optimization stage in order to incorporate energy storages into the final solution. The comparison of the iterative two-stage heuristic with commonly used approaches based on mixed integer linear programming shows outstanding results in terms of solution time and solution quality. Besides typically used test instances, the heuristic is applied to comprehensive case studies of the German electricity market, where (near-) optimal solutions can be derived for a yearly planning horizon with hourly time steps with computational effort of a few minutes using a standard PC.

1 Introduction and Problem Specification

Optimization models for electricity systems principally address a wide range of decision-making processes that, e.g., affect the fields of commodity trading and hedging, operation scheduling and maintenance control as well as portfolio and grid optimization. For generating companies, market or transmission operators and for the political world one of the most important and best known optimization problems is the so-called *unit commitment problem* (UCP) (e.g., [8]). Considering all types of electricity producing units, the aim of the UCP is to derive a feasible production schedule over an prescribed planning horizon (e.g., from real-time observations to long-term analysis) generally with minimal total operating costs. Within the opti-

A. Franz (✉) · J. Zimmermann
Operations Research Group, Institute of Management and Economics,
Clausthal University of Technology, 38678 Clausthal-Zellerfeld, Germany
e-mail: alexander.franz@tu-clausthal.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_21

mization, several techno-economic plant specific restrictions and system-wide constraints (e.g., a steady equilibrium between electricity provision and consumption) have to be considered in order to obtain a feasible schedule.

Ongoing changes in the regulatory environment (e.g., the transition from regulated to liberalized markets), in production technology, and in optimization techniques have been and still are key issues for the UCP for researchers and market participants all over the world. The latest changes refer in particular to the increasing share of prioritized renewable feed-in, which lead to major challenges in terms of (residual) demand coverage due to a significantly higher volatility and stochasticity. Consequently, the need for flexible, but cost-efficient power plant operations arises moving to the concept of load-following (instead of static base-load operations). The request for flexibility is usually supported by energy storage activities ensuring a smoothing of the volatile residual demand (demand minus intermittent renewable feed-in of e.g., wind and solar). The dispatching and scheduling of both, thermal plants and (mainly hydro-) storages, result in the UCP with *hydro-thermal coordination* (UCP-HT) [8].

The UCP and the UCP-HT are typically formulated as mixed-integer linear programs (MILP) [4], [8] where continuous decision variables determine the production level and binary decision variables the plant's status (on/off). Since the number of binary decision variables as well as the number of constraints in the model depend on the length of the planning horizon, only instances with a short-term planning horizon (i.e., 1 day or 1 week) can be solved to optimality within reasonable time. For the long-term instances focused in this article (e.g., 1 year), heuristics are necessary, in particular if a huge amount of different renewable-driven scenarios have to be calculated in reasonable time. According to the survey in [5], heuristics for the family of UC problems are divided into conventional techniques and metaheuristic algorithms. Conventional techniques include simple priority list-based methods, Dynamic Programming and decomposition techniques like Lagrangian Relaxation (e.g., [8]). Metaheuristics are often based on local search, genetic algorithms or simulated annealing (e.g., [3, 6]).

In what follows, we apply and evaluate a decomposition approach that splits the UCP-HT in an optimization of only thermal plant capacities on the first stage and a scheduling and coordinating of storages on the second stage. On both stages the residual demand is covered. Within the decomposition, a sequential use of an enhanced priority rule-based method, a repair procedure (ensuring solution feasibility), local search based improvement steps, and a demand-shifting process is used. These steps allow the tackling of well-known test instances with a large number of generating units, a long-term planning horizon, and hourly time-steps in a short amount of computation time. In order to evaluate the suitability for large-scale practical applications we also introduce a comprehensive case study of the German electricity market obtaining very convincing results by applying the proposed heuristic.

2 Model and Heuristic Solution Approach

Basically, the UCP-HT consists in finding an optimal production schedule for each thermal plant $i \in I$ and an optimal allocation of each (hydro) storage $j \in J$ to fulfill in particular a given customer demand at minimized production costs. The production schedule for each plant comprises decision variables for the binary on/off status and the continuous generation level of plant i for each point in time $t \in T$. The overall production costs due to the production schedule form the objective function:

$$\text{minimize} \quad TC = \sum_{t \in T} \sum_{i \in I} \left(FC_{it} + SC_{it} \right) + \sum_{t \in T} \sum_{j \in J} HC_{jt} + \sum_{t \in T} c^{\text{non}} N_t \qquad (1)$$

The total system operating costs TC covers relevant production-related components, consisting of thermal production costs FC_{it}, thermal start-up costs SC_{it}, (hydro) storage operating costs HC_{jt} as well as non-served energy costs $c^{\text{non}} N_t$. Thermal production costs are mainly dominated by fuel consumption and emission certificates and can typically be considered as the major cost component. Storage operating costs can often be neglected since no (direct) fuel costs occur for pumping or generating. Costs incurred by non-served energy penalize a (negative) deviation from the given energy demand in the amount of N_t and help to decrease computation time.

In addition to the (residual) demand coverage restriction, system-security constraints and production or supply-side constraints ensure the feasibility of an obtained solution. Whereas system-security constraints create the frame for a stable and secure operation (e.g., by compensating load imbalances utilizing reserve power), the production or supply side restrictions ensure a proper utilization within economic and technical specifications individually for each unit of each power source. In case of a thermal power plant, it should be guaranteed that, once a thermal generator is started (decommitted), it has to be online (offline) for at least its minimum up-time (minimum down-time). For (hydro) storages, energy balance and energy flow conservation equations are to be considered to calculate the amount of energy retained in each storage at each time t. Furthermore, logical constraints are necessary to determine the binary status of a shutdown or start-up of a thermal plant and to differentiate between e.g., a cold or a hot start-up of a plant.

All sketched constraints can be formulated as linear constraints which make the presented model to a MILP with a linearized production function. A detail review of a basic model (without energy storages) can be found in [1, 4] or [8]. For small- and mid-sized instances the resulting MILP model can be given to a solver (e.g., CPLEX) in order to obtain a solution schedule. In case of large-scale case studies and time-sensitive (e.g., scenario-based) analysis the *two-stage heuristic procedure* is recommended to use, because the UCP-HT is an NP-hard optimization problem.

The idea of the heuristic decomposition is the observation that the demand coverage constraint can already be satisfied by an isolated scheduling of only thermal plants (if sufficient thermal capacity is assumed). Therefore, a feasible solution of the UCP-HT can be found, although the possibility of demand-shifting through the

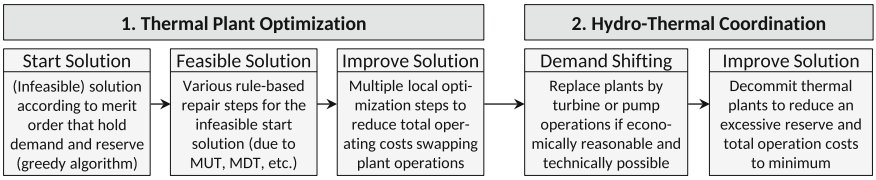


Fig. 1 Principal concept of the considered two-stage heuristic approach for the UCP-HT

set of hydro storages is neglected initially. Afterwards, within the second stage, hydro storages are stepwise embedded achieving a better objective function value. The resulting basic concept is illustrated in Fig. 1.

The *Thermal Plant Optimization* stage preselects certain plants to fulfill the volatile residual demand and spinning reserve requirements without using any storages. Moreover, several techno-economic parameters like power output specifications, minimum up-times (MUT) and minimum down-times (MDT) or time-dependent start-up costs are observed. In particular, the following three steps are determined: (i) create start solution by applying a greedy algorithm, (ii) deduce feasible solution via rule-based repair steps, and finally (iii) improve solution by local search. The Thermal Plant Optimization is followed by a re-optimization stage that iteratively replaces thermal plant operations by hydro storage activities in order to obtain lower total operating costs *TC*. Within the *Hydro-Thermal Coordination* stage a demand-shifting operation comprises not only a generation of electricity by storages (in high demand hours), but also a set of retaining phases to balance energy withdrawals (in low demand hours). The following local search improvement step enhances the solution by reducing redundant thermal plant operations and an excessive reserve provision by replacing plants by a set of other thermal plant activities.

Finally the solution schedule consists of an hourly dispatch of all thermal plants and all hydro storages that fulfills all constraints described above. A detailed explanation of the proposed two-stage heuristic procedure can be found in [2].

3 Numerical Results

Within our performance analysis, we compared the results of the two-stage heuristic procedure to the results obtained by CPLEX using an MILP formulation (similar to [4], enhanced by storages). Here, we present only an extract of our numerical study consisting of 100 problem instances classified in two test sets. Test set T_1 was first introduced by [3] and is commonly used for analyzing UCP models (e.g., in [4] or [6]). All instances consist of ten basic thermal units which are exactly replicated in order to get a $20, 30, \ldots, 100$ plant system. Thereby, the basic 24 h demand is adjusted relatively to the total capacity of the replicated system. For testing long-term planning horizons the 24 h planning horizon is copied to get instances of 1, 4,

Table 1 Computational results for T_1- and T_2-instances (avg. gap (max. gap) [%], avg. t^{cpu} [s])

Time horizon	#plants	MILP (T_1)		Heuristic (T_1)		MILP (T_2)		Heuristic (T_2)	
		Gap	t^{cpu}	Gap	t^{cpu}	Gap	t^{cpu}	Gap	t^{cpu}
1 week	10 – 100	0.2 (0.8)	11	0.8 (0.9)	0.1	0.3 (0.6)	31	0.9 (1.0)	0.1
4 weeks	10 – 100	0.5 (0.9)	177	0.8 (0.9)	0.5	0.6 (1.0)	198	0.6 (0.8)	0.3
12 weeks	10 – 100	0.7 (0.9)	558	0.9 (1.0)	4.2	0.5 (0.9)	3,406	0.7 (0.9)	2.4
20 weeks	10 – 100	0.6 (0.9)	1,103	0.9 (1.0)	11.0	7.4 (68.9)	3,586	0.8 (1.0)	5.8
1 year	10 – 100	6.3 (48.4)	14,245	**0.9 (1.0)**	**72.1**	62.8 (99.8)	14,167	**0.9 (1.0)**	**58.9**

Performed on an Intel Core i7-2760QM CPU with 2.7 GHz and 8 GB of RAM; GAMS 24.0 and CPLEX 12.4 used for MILP solutions (duality gap: 1%, time-limit: 5 h)

12, and 20 weeks as well as 1 year. In T_2, the recurrent demand pattern is replaced by the actual residual demand in Germany in 2012 (adjusted to the thermal capacity). Furthermore, in each instance of T_1 and T_2 hydro storages are added according to the total thermal capacity (about 7%, which is the current situation in Germany).

The average results of the 10–100 plant system are presented for each time horizon in Table 1, where worst-case gap-values are given in brackets. Columns 3–6 refer to T_1 and columns 7–10 to T_2. It can be observed that the presented two-stage heuristic delivers the best solution quality for long-term instances, whereas the solution time stands out and is always substantially lower compared to the MILP. Using a multi-start scheme with 100 iterations further gap-improvements of 0.1% are achievable. Consequently, our two-stage heuristic approach can be classified as a fast method for the UCP-HT which solves problems to near-optimality for all tested instances (according to the worst-case values no solution exceeds the 1% gap).

4 Case Study: German Electricity Market

The numerical results obtained in Sect. 3 make the proposed heuristic procedure highly suitable for the application of real-world energy systems, which we tested for the electricity system in Germany. Our case study comprises 216 instances that are spread across different scaling levels ranging from 25 to 500% of the German electricity market in order to extensively assess the capability of the proposed heuristic. Hence, the total thermal capacity $\sum P_i^{max}$ [GW], the yearly demand $\sum D_t$ [TWh], and the renewable feed-in $\sum wind_t$ and $\sum pv_t$ [TWh] are adjusted according to the scaling level based on the market structure in Germany in 2015 (cf. Table 2). For each instance the thermal plant portfolio with individually modelled large-scale units is fitted by applying a scaling method, so the portfolio composition is always comparable to the unscaled electricity system of Germany. The portfolio of hydro storages comprises only wholesale market participating storages (for the basic portfolio 6.5 GW)

Table 2 Computational results for the case study

#instances	Scaling	Time horizon (year)	#plants	$\sum_i P_i^{\max}$	$\sum_t D_t$	$\sum_t wind_t$	$\sum_t pv_t$	t^{cpu}
27	25	1	80	22.8	138.1	22.0	9.6	23.5
27	50	1	148	45.6	276.2	44.0	19.2	52.7
27	75	1	221	68.4	414.3	66.0	28.8	28.8
27	**100**	**1**	**289**	**91.3**	**552.4**	**88.0**	**38.4**	**150.6**
27	200	1	595	182.5	1104.8	176.0	76.8	369.0
27	300	1	870	273.8	1657.2	264.0	115.2	805.8
27	500	1	1467	458.4	2762.1	440.0	192.0	1711.5

and is scaled accordingly. For the creation of realistic hourly profiles of the demand and the renewable feed-in the methodology in [7] is applied. Utilizing the stochastic Ornstein-Uhlenbeck process presented by Wagner for residual demand modeling three scenarios, respectively for demand, wind, and solar are generated with calibration data from 2010 to 2015 (cf. ENTSO-E and German TSOs). Consequently, 27 possible instance combinations are introduced for each scaling.

The obtained results in Table 2 are averaged over all instances for each scaling level. As expected, low scaled instances are solved quite fast, but even for the entire German electricity system (scaling: 100%) near-optimal solutions can be derived in about 2.5 min of computation time. Moreover, a feasible hourly production schedule for an electricity system with a scaling of 500% (with almost 1,500 plants and in terms of the demand nearly comparable to the system of the EU-28) is received in less than half an hour. Therefore, it can be concluded that the two-stage heuristic procedure provides not only outstanding results for theoretical test instances, but is also excellent suited for comprehensive practical purposes.

5 Conclusion

With regard to the need of fast scheduling procedures for the operational planning of thermal power plants, renewables, and storages, an introduction to our two-stage heuristic for the UCP-HT was proposed within this paper. According to our numerical study, the decomposition method significantly outperforms MILP-based approaches for mid- and long-term planning horizons. Comparable outstanding performances can be observed for real-world case studies where the UCP-HT for the German electricity market is solved within minutes for 1 year. Future work may enhance the heuristic in terms of plant availabilities and demand side management.

References

1. Carrión, M., Arroyo, J.M.: A computationally efficient mixed-integer linear formulation for the thermal unit commitment problem. IEEE Trans. Power Syst. **21**, 1371–1378 (2006)
2. Franz, A., Rieck, J., Zimmermann, J.: Two-stage heuristic approach for solving the long-term unit commitment problem with hydro-thermal coordination. In: Operations Research Proceedings (2015)
3. Kazarlis, S.A., Bakirtzis, J.M., Petridis, V.: A genetic algorithm solution to the unit commitment problem. IEEE Trans. Power Syst. **11**, 83–92 (1996)
4. Morales-España, G., Latorre, J.M., Ramos, A.: Tight and compact MILP formulation for the thermal unit commitment problem. IEEE Trans. Power Syst. **28**, 4897–4908 (2013)
5. Saravanan, B., Das, S., Sikri, S., Kothari, D.P.: A solution to the unit commitment problem—a review. Front. Energ. **7**, 223–236 (2013)
6. Viana, A., Pinho de Sousa, J., Matos, M.A.: Fast solutions for UC problems by a new metaheuristic approach. Electric Power Syst. Res. **78**, 1385–1395 (2008)
7. Wagner, A.: Residual demand modeling and application to electricity pricing. Energ. J. **35**, 45–73 (2014)
8. Wood, A.J., Wollenberg, B.F., Sheblé, G.B.: Power Generation, Operation, and Control. Wiley, Hoboken (2014)

Flexibility Options for Lignite-Fired Power Plants: A Real Options Approach

Barbara Glensk and Reinhard Madlener

Abstract Germany's energy system transformation process "Energiewende" implies, on the one hand, the promotion of renewable energy sources and, on the other hand, difficulties in the profitable operation of many modern conventional power plants due to increasing shares of renewable electricity and decreasing electricity wholesale prices. Nevertheless, the prioritized conventional power generation technologies are still needed in times of low wind and solar power generation in order to maintain the security of electricity supply. Regarding these aspects and the specific situation in the federal state of North Rhine-Westphalia, the problem of further operation of lignite-fired power plants is of particular importance. In the study undertaken we tackled the following research questions: Should lignite-fired power plants be operated without any changes until the end of their lifetime? Can already existing lignite-fired power plants be operated more flexibly? If so, which flexibility options should be taken into consideration? What is the optimal investment time for these flexibility options? Are investments in other power generation technologies more suitable for ensuring system stability than investing in the retrofitting of existing lignite-fired power plants is? To answer these questions, we propose an optimization model that is based on real options analysis (ROA) and, more precisely, on the option of choosing. In the proposed model, the economic as well as technical aspects of the power plant operation are taken into consideration for the profitability calculations. Moreover, the results show the importance of the subsidies for lignite-fired power plants and their further operation.

B. Glensk (✉) · R. Madlener
School of Business and Economics, Institute for Future Energy Consumer Needs and Behavior (FCN), E.ON Energy Research Center, RWTH Aachen University, Mathieustrasse 10, 52074 Aachen, Germany
e-mail: BGlensk@eonerc.rwth-aachen.de

R. Madlener
e-mail: RMadlener@eonerc.rwth-aachen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_22

1 Introduction

The main challenge of Germany's energy system transformation process is to develop a future electricity market that meets three main goals: to ensure security of supply, to limit the costs, and to enable innovation and sustainability, when a large share of the power is derived from intermittent renewable energy sources. To address this challenge, the existing electricity market is to be converted into an "electricity market 2.0", where fossil-fueled fired power plants will take on a new role as partners of renewables (back-up capacities to cover the variability of net demand). The required back-up (i.e. in the role of reserve) capacities of conventional power plants in the "electricity market 2.0" are to be remunerated via the market mechanisms without strong interventions in the existing market design [2]. From this perspective, the flexible and efficient operation of the country's conventional power plants becomes more important.

Regarding the federal state of North Rhine-Westphalia, Germany's "Energieland" and "Industrieland" No. 1, where lignite- and hard-coal-fired power plants are the dominant power and heat generation technologies, their flexibility is key for the reliable operation of the whole power system and the security of supply for existing industries and households. The flexible operation of lignite-fired power plants in NRW, which according to [2] should represent back-up capacities in the "electricity market 2.0", is constrained by the technical restrictions of this technology, defined by ramping capability, minimum load, as well as must-run requirements [5].

The investigations of which flexibility option should be applied—if any at all—as well as when the flexibility option should be exercised, can be undertaken using real options analysis (ROA). The traditional now-or-never discounted cash flow analysis is no longer an adequate approach for this type of decision process, as it does not take managerial flexibility appropriately into account. Real options valuation is based on option pricing methods used in finance, developed by Black, Scholes and Merton (see e.g. [1] or [12]), and extended by Dixit and Pindyck [4] and others to real assets. In the meantime, ROA has been applied to many different industries (see e.g. [13] or [16]). For the energy sector in particular, a comprehensive review of the literature is provided by Fernandez et al. [6].

In our study, we consider the situation of an already existing lignite-fired power plant, and some possible investor decisions to be undertaken: (1) to continue the operation of the existing power plant; (2) to abandon this activity; or (3) to invest in flexibility options (retrofitting measures). In order to solve this decision-making problem, we propose the application of the option of choosing, which is a combination of multiple other options.

2 Model Specification

The proposed model, which is based on the real options approach, is not a simple set of equations but rather a procedure which supports the decision-making process. This procedure consists of several steps. First, the operation strategy for each hour of the power plant operation is defined, on which the cash flows and project values of the power generation can be estimated. This step is conducted based on the methodology proposed by Glensk and Madlener [7]. By the definition of the operation strategy, the spark spread (i.e. the difference between the electricity price and the fuel price regarding the load-level-dependent net efficiency factor) is used as the profitability indicator and source of uncertainty, and estimated via the arithmetic Brownian motion process.[1] Using this approach, the expected project value ($E(PV)$) for an existing power plant with and without a retrofit measure (i.e. some technical element which can improve the flexible operation of the power plant) can be calculated and used in a further ROA.

In the next step, the binomial lattice for the option of choosing between continuation, abandonment, and expansion can be applied. The binomial lattice method is one of several real options solution approaches (such as partial differential equations with a closed-form model or simulations, etc.). The major advantage of this method is, on the one hand, its ease of use and better tractability; on the other hand, it allows a flexible use of different types of real option problems. Especially the discussed option of choosing is an American-style option (i.e. it can be exercised at any time) compared to which the closed-form model (which allows only one exercise date) is inadequate. Applying the binomial lattice approach we specify:

(1) The lattice of the present value (PV_t) of the underlying asset, i.e. how the underlying asset, in our case the project's present value obtained from the first step, changes over time. Based on the assumed normal distribution of the underlying asset, the "up" and "down" movement parameters are defined as follows:

$$up = e^{(\sigma\sqrt{\Delta t})} \quad \text{and} \quad down = e^{(-\sigma\sqrt{\Delta t})} \tag{1}$$

where σ is the associated volatility, and Δt is the incremental time.

(2) The option valuation lattice using backward induction. Beginning at the last year ($t = T$) on the lattice of the underlying asset, the maximum value of the continuation value, abandonment value, and expansion value, respectively, can be chosen. The continuation value (CV_t), abandonment value (AV_t), and expansion value (EV_t) are determined as follows:

$$CV_t = \begin{cases} PV_t \quad \text{for} \quad t = T \\ PV_t = \frac{prob \cdot PV_{t+1,up} + (1-prob) \cdot PV_{t+1,down}}{e^{rf \cdot \Delta t}} \quad \text{for} \quad t = T-1, T-2, \ldots, 0 \end{cases} \tag{2}$$

$$AV_t = 0.05 \cdot InvCosts \tag{3}$$

[1] For more information, see [7].

$$EV_t = Expansion\,factor \cdot PV_t - Expansion\,costs \tag{4}$$

where $PV_{t+1,up}$ and $PV_{t+1,down}$ are the project's present values after "up" and "down" movement in the subsequent time period $t + 1$, respectively, rf is an risk-free rate, and *prob* denotes the risk-neutral probability, given as:

$$prob = \frac{K - down}{up - down} \tag{5}$$

with K as risk-adjusted growth factor of the underlying asset.[2]

3 Case Study

In our case study, we applied the proposed approach to the lignite-fired power plant Goldenberg in North Rhine-Westphalia,[3] which is owned by RWE and was commissioned in 1993 with a net installed capacity of 171 MW. Its net thermal efficiency is ca. 40% and it is used for baseload power generation for private households as well as industry (paper and chemicals industry) [15].

All techno-economic parameters of the considered power plant that are necessary for calculating the power plant's value can be found in [9].

To make existing lignite-fired power plants more flexible, different flexibility options are possible. These flexibility options can be subdivided into different groups, such as spatial flexibility options (regarding the electricity distribution network), storage, or timed flexibility options (for the supply and the demand side) [3]. Nevertheless, the most frequently used flexibility option for the supply side is the retrofitting option of an existing power plant. In our case study, we consider the retrofit measure for the firing system of the lignite-fired power plant. This technical component enables the reduction of the minimum load of the power plant and the increase of its present value (about 12% for minimum load of 50% and about 25% for the minimum load of 40%). Unfortunately, reinvestment in the firing system is connected with some costs. In our case, these amount to about 30% of the total new investment costs [9, 14].

The results regarding the retrofit measure which decreases the minimum load level from 60 to 50% are presented in Table 1. Notice that according to our results the lignite-fired power plant should be operated without any extension until the end of its lifetime. In Table 1 the positive development of the present value of the power

[2] The use of this factor is adequate for non-traded underlying assets (cf. [10]).

[3] It is one of the lignite-fired power plants considered in the project "Verbundprojekt Transformationsprozesse für nachhaltige und wettbewerbsfähige Wirtschafts- und Industriestrukturen in NRW im Kontext der Energiewende"—Virtuelles Institut "Transformation—Energiewende NRW" (for more information see [9]).

Table 1 Binomial lattice when the minimum load-level is 50% and the support subsidies are included (UA—value of underlying asset [in €], OV—option value [in €] and D—decision)

0	1	2	…	14	15
UA = 28,461,422 OV = 14,080,566 D = continue	UA = 28,556,278 OV = 14,806,193 D = continue	UA = 28,651,630 OV = 15,569,215 D = continue	…, …, …,	UA = 29,820,990 OV = 28,454,206 D = continue	UA = 29,920,565 OV = 29,920,565 D = continue
	UA = 28,366,526 OV = 14,707,808 D = continue	UA = 28,461,244 OV = 15,465,760 D = continue	…, …, …,	UA = 29,622,834 OV = 28,265,132 D = continue	UA = 29,721,747 OV = 29,721,747 D = continue
		UA = 28,272,123 OV = 15,362,992 D = continue	…, …, …,	UA = 29,425,994 OV = 28,077,314 D = continue	UA = 29,524,250 OV = 29,521,250 D = continue
			…,	…,	…,
				UA = 27,163,498 OV = 25,918,514 D = continue	UA = 27,254,199 OV = 27,254,199 D = continue
					UA = 27,073,098 OV = 27,073,098 D = continue

plant (i.e. UA—values of the underlaying asset) can be observed (the same results are obtained by decreasing the minimum load level from 60 to 40%). Nevertheless, it should be noted that the made calculations take the subsidies for lignite power plants into account [11]. Without subsidies, the present value of the operated power plant are negative, even when the same retrofit measure is considered.[4] In such a situation the further operation of the power plant is no longer economical and thus reasonable, and ought to be stopped.

4 Conclusions

The increased use of renewable energy technologies for electric power generation and the existing support schemes for renewables have a significant impact on the merit order of power plant dispatch as well as the electricity price. The owners of conventional power plants such as lignite-fired ones, which were designed as baseload technologies, face severe changes to their operating strategies (i.e. decreasing number of operation hours as well as increasing number of breaks). Moreover, following the concept of the "electricity market 2.0", the conventional power generation tech-

[4]For more results, see [9].

nologies will be compelled to optimize their operations and make them more flexible. The procedure proposed in this paper, and which is based on the real options methodology, constitutes a useful tool for the decision-making process. First, the procedure allows the determining of the simplified operation strategy for the power plants, and shows the expected future role of conventional power generation as back-up capacities (because of interrupted operation, electricity delivered on demand, and more shut-downs and start-ups). Second, using the real options approach, market uncertainties (such as the stochastic development of electricity, fuel or CO_2 prices) can be easily incorporated into the model structure and can positively impact the results. Third, changes in the values of some of the model parameters, such as the subsidy level, show direct implications of policies and policy changes for market participants and their optimal decision-making.

Further investigations and analysis of different power plants (considered in the underlying project), as well as different flexibility measures, are planned in order to check the robustness of the model results. Furthermore, a sensitivity analysis, especially with regard to subsidies policy, will be undertaken.

References

1. Black, F., Scholes, M.: The pricing of options and corporate liabilities. J. Polit. Econ. **81**(3), 637–654 (1973)
2. BMWi: Ein Strommarkt für die Energiewende - Ergebnispapier des Bundesministeriums für Wirtschaft und Energie (Weißbuch). BMWi, Berlin (2015)
3. Connect: Leitstudie Strommarkt Arbeitspaket Optimierung des Strommarktdesigns. Connect energy economics GmbH. Berlin (2014)
4. Dixit, A.K., Pindyck, R.S.: Investment under Uncertainty. Princeton University Press, Princeton (1994)
5. Ecofys: Flexibility options in electricity systems. Ecofys, Berlin (2014)
6. Fernandez, B., Cunha, J., Ferreira, P.: The use of real options approach in energy sector investments. Renew. Sustain. Energ. Rev. **15**, 4491–4497 (2011)
7. Glensk, B., Madlener, R.: A real options analysis of the flexible operation of an enhanced gas-fired power plant, FCN Working Paper No. 11/2015. RWTH Aachen University (2015)
8. Glensk, B., Madlener, R.: Investments in flexibility measures for gas-fired power plants: a real options approach. In: Drner, K.F., Ljubic, I., Pflug, G., Tragler, G. (eds.) Operations Research Proceedings 2015: Selected Papers of the International Conference of the German, Austrian and Swiss Operations Research Societies, 1–4 Sept 2015, Vienna, Austria. Springer, Berlin (in press) (2016). ISBN 978-3-319-42901-4
9. Glensk, B., Madlener, R.: Evaluating the enhanced flexibility of lignite-fired power plants: a real options analysis. FCN Working Paper No. 10/2016. RWTH Aachen University (2016)
10. Guthrie, G.: Real Options in Theory and Practice. Oxford University Press, New York (2009)
11. Küchler, S., Meyer, B.: Billiger Strom aus Atom und Kohle? Staatliche Förderungen 1970–2008. Forum Ökologisch-Soziale Marktwirtschaft. Berlin (2010)
12. Merton, R.C.: Theory of rational option pricing. Bell J. Econ. Manag. Sci. **4**(1), 141–183 (1973)
13. Mun, J.: Real Options Analysis: Tools and Techniques for Valuing Strategic Investment and Decisions. Wiley, Hoboken, New Jersey (2006)
14. Plewnia, M.: Impact of flexibility measures on conventional power plants. Master's Thesis, Institute for Future Energy Consumer Needs and Behavior (FCN), RWTH Aachen University, Aachen, Germany (2014)

15. Goldenberg, R.W.E.: http://www.rwe.com/web/cms/de/60098/rwe-power-ag/energietraeger/braunkohle/standorte/edz-kw-goldenberg/ (2016). Accessed 4 Aug 2016
16. Trigeorgis, L.: Real Options: Managerial Flexibility and Strategy in Resource Allocation, 5th edn. The MIT Press, Cambridge, Massachusetts (US)/London, England (UK) (2000)

Needmining: Evaluating a Whitelist-Based Assignment Method to Quantify Customer Needs from Micro Blog Data

Niklas Kuehl and Marc Goutier

Abstract In the paper at hand we evaluate how a basic whitelist-approach with keywords performs on automatically assigning micro blog data (tweets) to customer need categories in the field of e-mobility. We are able to identify certain characteristics that determine the classification success like unambiguousness and uniqueness of the whitelist words.

1 Introduction

The identification of customer needs in early design stages of new services and products is an important task, which is addressed among different disciplines. A new approach called *Needmining* evaluates the feasibility of automatically identifying customer needs from micro blog data [2]. As a first case study, we used a Twitter data set of 2400 German tweets from 2015 in the field of e-mobility. In a previous paper, we showed the feasibility to classify tweets on whether or not they contain customer needs [4]. After successfully identifying these "need tweets", it is important to identify the needs themselves. In this paper, we aim to quantify previously known needs from a literature review in the Twitter data set (see Fig. 1) and evaluate the performance of the automatable approach. For a previous categorization of e-mobility needs, we use four major need categories as presented in [3].

N. Kuehl (✉) · M. Goutier
Karlsruhe Service Research Institute (KSRI), Karlsruhe Institute of Technology (KIT), Englerstr. 11, 76131 Karlsruhe, Germany
e-mail: kuehl@kit.edu

M. Goutier
e-mail: marc.goutier@student.kit.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_23

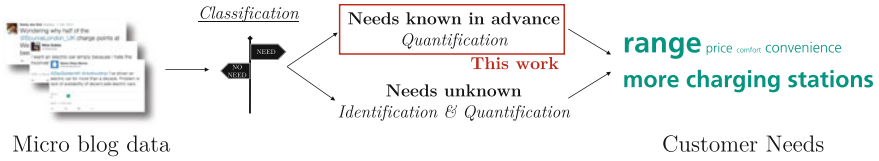


Fig. 1 General needmining approach and relation to this work

Table 1 Amount of words per whitelist

Need	Whitelist				
	① Original	② Manually stemmed	③ Synonyms	④ Thesaurus	⑤ Similarity
Cost	37	16	160	153	1580
Car	35	29	536	618	2557
Charging	21	7	79	88	592
Social	22	21	219	353	1812
Total	115	73	994	1212	6541

2 Method

The proposed method is independent of a specific domain, but is shown exemplary in the domain of e-mobility. We assign every instance (=single tweet) in our dataset to need categories, which are represented by whitelists containing need expressions in the form of single words.

In [3], we analyzed the current state of research in the field of e-mobility. As an outcome, we were able to identify four major need categories, namely *cost-related*, *car-related*, *charging-related* and *social and individual* needs. We create a list for every major category and fill these lists with the need expressions from the examined publications ①. Since the terminology of the publications is—opposed to our dataset—largely English, we have to translate the need keywords into their corresponding German term(s). The amount of words per whitelist is depicted in Table 1.

Since a particular need can be expressed in different words, we also implement the option to enrich the lists with thesauri of the (need) words by leveraging the "Wortschatz-Portal" [5]. It allows to call three different functions: *Synonyms* returns thesauri for a given input word ③. *Thesaurus* additionally returns thesauri for the lemmatized input word ④. *Similarity* returns every other word which is related to the input word in some kind (like antonyms, hyperonyms, cohyponyms and other) ⑤.

To assign instances of our dataset we look at every instance separately and compare the letter sequence of the words in the whitelists with the letter sequences of the instance. If one word of one category also occurs in the instance, we assign the instance to this category. Multiple assignments of one instance to several major need categories are allowed. Moreover, we assign instances to the *other* category, when

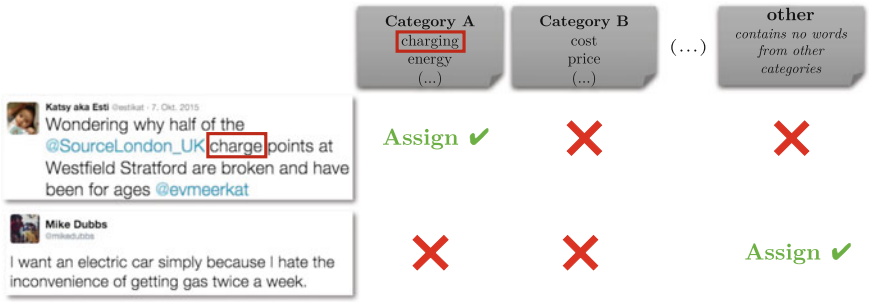


Fig. 2 Exemplary assignment of two instances and two major need categories

the content does not match with any of the words in our categories (see Fig. 2). The quantification of one need is the amount of instances which are assigned to the corresponding category.

Additionally, we implement the possibility to stem the expressions of the need categories as well as the content of the instances by using the German Stemmer [1]. Since a stemmer also unifies words, e.g. substitutes every umlaut with the corresponding vowel, it is only useful to stem both the content and the words in the need category whitelists—or none. We perform the quantification for every whitelist we generated (original whitelists, enriched with the function *Synonyms*, *Thesaurus* or *Similarity*) and the corresponding form with stemmed words. In addition to the eight resulting possibilities, we also stem the original category lists manually to compare the results of manual and automated stemming. The use of these new lists and a version which is stemmed manually as well as by the German Stemmer add two other possibilities to our results.

3 Results

To evaluate our approach to assign instances form the previous chapter, we require a benchmark. We select the labeled sample from [3], in which we manually assigned every instance to one or more major need category or to the *other* category. Although our approach is not supervised, we are able to use assessment schemes from supervised statistical learning classification. When we compare the assignment of our model based on the whitelists and the manual allocation, a single instance can be in four conditions when we regard their assignment to one major category. In case an instance is assigned by the model to the same category like in the manual allocation, the assignment is "correct" and it is either a true positive or a true negative (depending on the instance belonging to the regarded major category). In case the method assigns an instance to a different category to which the instance does not belong, the state is called a false positive. By contrast, a false negative occurs when

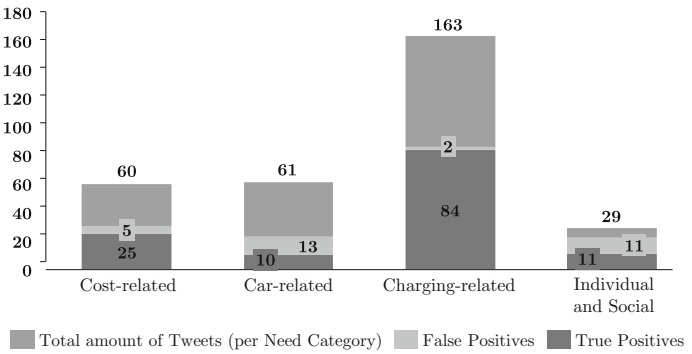


Fig. 3 Comparison of the manual and the automatic assignment. Computer assignement used original whitelists. Lists and instances are stemmed

an instance is assigned to a category only by the manual allocation and not by the automated method as well. We conduct the evaluation for every instance and every major need category. Regarding every major need category individually, we get the total number of true positives, true negatives, false positives and false negatives. These total numbers allow us to calculate the performance indicators precision and recall. These indicators help us to analyze the performance of the assignment in total. Figure 3 depicts the number of tweets which are assigned to the major need categories in case we use the original whitelists and perform stemming. Table 2 displays the performance of every version of quantifications we implemented and executed.

Regarding the original need categories without any enrichments ①, the precision is fundamentally better than a random guess. Even the lowest precision in the category *car* achieves 43% (stemmed)[1] and 45% (unstemmed). The major categories *cost* and *charging* show good precision performances above 80%. This is reasoned in the special manner of these categories: They contain unique words in their whitelists, namley "Preis" (price) in the *cost-related* category and "Ladestation" (charging station) and "Reichweite" (driving range) in the *charging* category. These words which are not only very frequent in their corresponding instance, but are also unambiguous and therefore usually only appear in the same context as in the corresponding literature. Consequently, there are also less false assignments of tweets which contain these words, but do not fit in either one of these two categories. When we observe the other two categories *car* as well as *social and individual*, they do not feature such unique words which could lead to comparable precision values. Referring the *social and individual* category, this insight could be surprising, because "Umwelt" (environment) has the same attributes to be seen as a unique and common word. However, the expression "environmental" is not used very often in *social and individual* tweets and for that reason, it does not increase the precision significantly. Our performance is different regarding the recall indicator. Its results are poor, only the *charging-related* category achieves a value above 50% (stemmed).

[1] In general, the repercussions of automatic stemming are nominal and are not discussed further.

Table 2 Results of assignment for every tested combination

need category	whitelist:	① original		② manually stemmed		③ *Synonyms*		④ *Thesaurus*		⑤ *Similarity*	
	stemmed:	no	yes	no	yes	no	yes	no	yes	no	yes
cost	recall	37%	42%	48%	47%	52%	62%	48%	58%	92%	93%
	precision	85%	83%	83%	85%	53%	39%	47%	38%	19%	18%
car	recall	15%	16%	13%	16%	67%	85%	84%	89%	100%	100%
	precision	45%	43%	50%	50%	17%	19%	18%	18%	18%	18%
charging	recall	36%	52%	71%	71%	42%	65%	40%	58%	72%	91%
	precision	94%	98%	95%	95%	85%	71%	93%	86%	63%	52%
social	recall	38%	38%	45%	45%	48%	66%	72%	86%	100%	100%
	precision	73%	50%	72%	50%	31%	15%	13%	13%	10%	9%
other	recall	93%	93%	89%	87%	26%	9%	9%	4%	0%	0%
	precision	20%	23%	26%	26%	17%	17%	14%	13%	0%	0%

Nevertheless, *cost* and *charging* also show a higher recall, especially compared to *car*, which is again originated in the unique words which cover a great amount of relevant instances. Additionally, also the character of the need categories play a role. Every major need category contains needs, but they do not contain their characteristic attributes. *Cost-related* or *charging-related* needs are predominantly expressed with the tangible need, e.g. "the price is too high", whereas especially *car-related* needs are expressed with their characteristics attributes "I want a red car" instead of "I want a car in a specific color". This phenomenon leads to the issue that these instance are not recognized by our method.

In summary, the recall is too low to project the automatically identified quantities of tweets as an estimation for the full dataset. To project the real quantities, we would also need a comparable precision in every category. Since the precision deviates drastically from category to category, a projection would only reinforce the different levels of precisions from the allocation. Nevertheless, it is interesting to note that the total amount of instances of a specific major need in the dataset has no influence on recall and precision. We can show this best when focusing on *cost-* and *car-related* need. Although the amount of both categories is about 60 respectively 61 instances, the *cost-related* category achieves results in recall and precision which are equal to the best-performing categories in our study, whereas the indicators of the *car-related* needs have the lowest numbers in almost every condition.

The manual stemmed lists ② contain generously stemmed words by a human. Since human stemming violates our principle of an automated and scalable implementation, this part only acts as a reference at which level our approach is able to perform with an almost perfect stemming procedure. Overall, the results are better compared to the original list, for both recall and precision.

The last three deviations from the original whitelists are iterations with enriched major need categories ③ ④ ⑤. As expected, the recall increases, compared to the original condition, when we add thesauri with the *Synonyms* function ③. Especially the recall for *car* raises dramatically. This observation is reasoned in the fact that this major need category covers a large ontological area, e.g. the word "motor" has plenty

of thesauri. This leads to the circumstance that the *car-related* needs encompass over 500 needs after the enrichment (c.f. Table 1), which obviously benefits the recall. On the contrary, the precision decreases drastically. Most of the added thesauri are too unspecific which causes many false assignments. Only the *charging-related* category remains on an acceptable level because it is narrowly specified in our study and therefore, there are less existing thesauri. Stemming enforces the described phenomenons on recall and precision. Consequently, it is not recommended. Implications for our quantification are the enrichment with thesauri based on the *Synonyms* function ③ increases the recall which is preferable, but, the loss of precision is so drastically, that even with the higher recall a projection of the total numbers in the original dataset is impossible.

The enrichment with the *Thesaurus* function ④ in our method or the addition of any kind of words which are related in some way with our needs (*Similarity* function ⑤) intensifies the findings of the previous paragraph. Major needs are not able to catch up by adding thesauri, neither absolute nor relative to the other categories.

4 Conclusion

We evaluated a basic whitelist approach to assign micro blog instances containing customer needs to one of four major need categories. Major limitations are the specificity of the data set and the manual effort still necessary to gain information on the need categories in the first place. Nonetheless, the results are already promosing for practitioners in some categories. From a theoretical perspective it is interesting to note that automatic stemming did not increase performance significantly. Important characteristics for classification are unambiguousness and uniqueness of the words in the whitelists. As a next step, it is of importance to automate the filling of the whitelists (e.g. by leveraging knowledge databases like Wikipedia) and improving the classifiction results further.

References

1. Caumanns, J.: GermanStemmer. https://lucene.apache.org/core/2_9_4/api/contrib-analyzers/org/apache/lucene/analysis/de/GermanStemmer.html. Last accessed 23 Apr 2016
2. Kuehl, N.: Needmining: towards analytical support for service design. In: Borangiu, T., Dragoicea, M., Nóvoa, H. (eds.) Exploring Services Science: 7th International Conference, IESS 2016, Bucharest, Romania, 25–27 May 2016, Proceedings, pp. 187–200. Springer International Publishing (2016)
3. Kuehl, N., Goutier, M.: "Need tweets": new insights about customer needs from micro blog data in the field of e-mobility. In: INFORMATIK 2016. Lecture Notes in Informatics (LNI) (2016)
4. Kuehl, N., Scheurenbrand, J., Satzger, G.: Needmining: identifying micro blog data containing customer needs. In: 24th European Conference of Information Systems (2016)
5. University, L.: Wortschatz-Portal. http://wortschatz.uni-leipzig.de/. Last accessed 23 Apr 2016

Optimising the Natural Gas Supply Portfolio of a Gas-Fired Power Producer

Nadine Kumbartzky

Abstract The expansion of gas-fired power plants has led to increasing interactions between the natural gas and electricity market. In order to effectively manage risk of volatile energy prices, operators of gas-fired power plants need to take natural gas procurement and power plant resource planning simultaneously into account. An industrial company is considered that owns a gas-fired combined heat and power (CHP) plant. To ensure a stable heat and power supply, the amount of gas needed to operate the CHP plant must be available at any time. Natural gas can be procured by signing supply contracts or by engaging in the natural gas spot market. A two-stage stochastic MILP model is proposed that optimises the gas supply portfolio and the CHP plant operation according to revenue potential in the electricity spot market. Uncertainty of gas and electricity spot prices is addressed by means of stochastic processes. Price risk is explicitly taken into account by the Conditional Value-at-Risk (CVaR). A convex combination of expected total costs and CVaR allows for representing different risk preferences of the decision maker. The efficient performance of the presented approach is illustrated in a case study using the example of German energy markets. The results reveal the significant influence of different risk preferences on the optimal gas supply portfolio composition.

1 Introduction

The availability of energy is a necessary prerequisite in the daily business of many industrial companies. For the production of commodities, a certain amount of power and process heat is needed. CHP plants simultaneously generate heat and power in a coupled process resulting in a high efficiency. In 2014, CHP plants accounted for 70.4% of industrial electricity generation and even 88.1% of industrial net heat generation in Germany, respectively [2]. The heat demand of the considered company can only be fulfilled by the CHP plant. By contrast, the power demand is either satisfied

N. Kumbartzky (✉)
Chair of Operations Research and Accounting, Faculty of Management and Economics, Ruhr University Bochum, Bochum, Germany
e-mail: nadine.kumbartzky@rub.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_24

by the CHP plant or by purchasing power from the electricity spot market. Anyhow, the gas needed to operate the CHP plant must be available at all times.

Natural gas can be procured by signing bilateral contracts or by purchasing gas from the spot market. Additionally, some companies have access to a gas storage facility. In order to procure gas at the lowest possible cost, the industrial company has to efficiently manage its gas supply portfolio. Optimising natural gas procurement has already been discussed in literature, e.g. in [1, 4]. However, gas procurement for gas-fired power plants must not neglect power plant resource planning. The reason is that decisions on gas procurement depend on the operating schedule of the CHP plant which in turn depends on the development of the electricity spot market. To the best of our knowledge, the problem of optimising the gas supply portfolio of an industrial company operating a gas-fired power plant has not been addressed so far. In this paper, a stochastic MILP model is proposed that simultaneously optimises gas procurement, CHP plant operation as well as trading in the electricity spot market.

2 Natural Gas Supply Options

The industrial company has basically two options to procure the natural gas needed to operate the CHP plant: either by signing bilateral contracts or by purchasing gas directly from the spot market. In the following, two types of bilateral contracts are considered: baseload and open contracts. Baseload contracts are characterised by a fixed contracted capacity that is consumed at a constant level throughout the contract period. Hereafter, baseload contracts are modelled as take-or-pay contracts with a take-or-pay level of 100%. Hence, the company is either required to procure the contractually determined cumulative capacity of gas or must pay for it even if the entire quantity of gas cannot be consumed [4]. On the contrary, the amount of gas purchased by open contracts can vary over time according to the specific needs of the industrial company. Due to this granted flexibility, purchase prices of open contracts are typically higher than those of baseload contracts. If the contract period is less than a year, then purchase prices of baseload and open contracts are typically fixed for the whole contract period. This provides the advantage that purchase costs can be identified in advance. However, the company cannot profit from decreasing market prices. Furthermore, purchase prices of bilateral contracts commonly contain a risk margin to compensate the supplier for bearing price risk.

Additionally, the industrial company can buy gas from the spot market or sell excess gas to the spot market. This creates flexibility and offers the possibility to benefit from decreasing market prices. However, gas spot prices are volatile and uncertain. Thus, the company possibly has to cope with increasing market prices when relying on the spot market. Moreover, the industrial company has the option to rent gas storage capacity which allows for decoupling gas procurement and gas consumption regarding time. In times of low market prices, excess gas can be injected

into the storage facility and withdrawn when prices rise again. Apart from a fixed storage fee (depending on the injection and extraction rates as well as on the working gas volume) variable costs need to be paid for the actual amount of gas in storage.

3 Two-Stage Stochastic Optimisation Model

A two-stage stochastic MILP model is used for the simultaneous optimisation of natural gas procurement and power plant resource planning. A time horizon of T time periods $t \in \mathscr{T} = \{1, \ldots, T\}$ is considered. Uncertain gas and electricity spot market prices are represented by a finite number of scenarios $s \in \mathscr{S}$. Let $\mathscr{J}$ be the set of baseload contracts and $\mathscr{K}$ be the set of open contracts. The objective is to minimise total costs C_s^{total} that consist of gas procurement costs of bilateral contracts, costs of trading in the gas and electricity spot market, storage costs as well as generation costs of the CHP plant (including start-up and shut-down costs).

An overview of the decision-making process is given in Fig. 1. At the beginning of the planning horizon, the company needs to decide if gas storage capacity is rented and which bilateral contracts to be signed. If a baseload contract is concluded, also the contracted capacity has to be settled. In response to the realisation of uncertain gas and electricity spot prices, the actual quantities procured by bilateral contracts as well as gas spot market purchases and sales are determined. If gas storage capacity is rented, then injection and withdrawal quantities are scheduled. Furthermore, decisions on the hourly power and heat generation, on the operating schedule of the CHP plant as well as on trading activities in the electricity spot market also belong to the second stage.

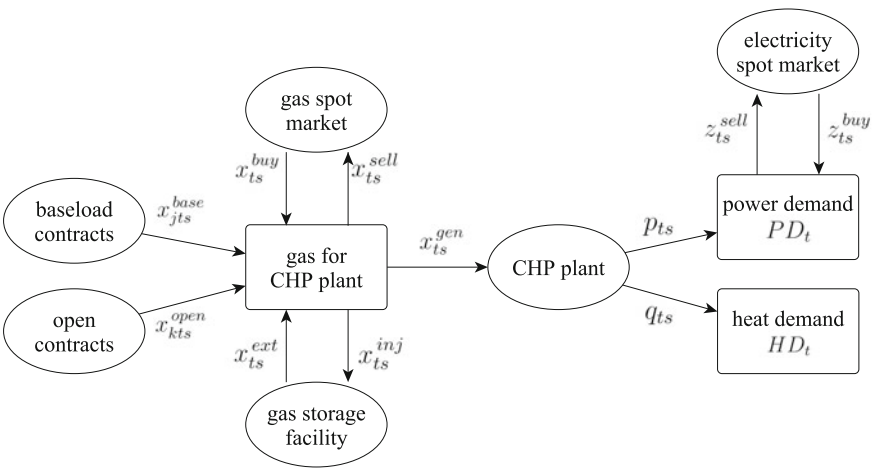


Fig. 1 Decision-making process of gas supply and CHP plant operation

In the following, we exemplarily present some parts of the MILP model. The minimisation of total costs C_s^{total} is subject to the following constraints:

$$PD_t = p_{ts} + z_{ts}^{buy} - z_{ts}^{sell} \qquad \forall\, t \in \mathscr{T}, \forall\, s \in \mathscr{S} \quad (1)$$

$$HD_t = q_{ts} \qquad \forall\, t \in \mathscr{T}, \forall\, s \in \mathscr{S} \quad (2)$$

$$x_{ts}^{gen} = \sum_{j \in \mathscr{J}} x_{jts}^{base} + \sum_{k \in \mathscr{K}} x_{kts}^{open} + x_{ts}^{buy} - x_{ts}^{sell} + x_{ts}^{ext} - x_{ts}^{inj} \qquad \forall\, t \in \mathscr{T}, \forall\, s \in \mathscr{S}. \quad (3)$$

Constraints (1) ensure that the power demand can either be satisfied by own generation or by participating in the electricity spot market, whereas constraints (2) model that the heat demand can only be fulfilled by the CHP plant. In (3), the amount of gas x_{ts}^{gen} needed to operate the CHP plant equals the quantity consumed by baseload and open contracts plus spot market purchases and extraction quantities minus spot market sales minus the amount of gas injected into the storage facility. The operation of the CHP plant is modelled as presented in [5] assuming that generation costs can be represented by a convex function of heat and power generation. Furthermore, additional constraints are introduced that model e.g. start-ups and shut-downs of the CHP plant, gas procurement by bilateral contracts and gas storage activities.

As a measure of price risk, the CVaR is utilised. The CVaR at confidence level α is defined as the expected costs given that the costs are greater or equal to the Value-at-Risk ζ at confidence level α [6]. The authors also present a mathematical formulation for minimising the CVaR which is used below. In order to represent different risk preferences of the decision maker, a convex combination of expected total costs and CVaR is minimised, i.e.

$$\min \quad (1-\beta) \sum_{s \in \mathscr{S}} \pi_s\, C_s^{total} + \beta \left(\zeta + \frac{1}{1-\alpha} \sum_{s \in \mathscr{S}} \pi_s\, h_s \right) \quad (4)$$

$$\text{s.t.} \quad (1) - (3) \quad \& \text{ additional constraints}$$

$$C_s^{total} - \zeta - h_s \leq 0 \quad \forall\, s \in \mathscr{S} \quad (5)$$

$$h_s \geq 0 \quad \forall\, s \in \mathscr{S}, \quad (6)$$

with auxiliary variables h_s, π_s denoting the probability of scenario s, and $\beta \in [0, 1]$ being a weighting factor. The convex combination allows for representing different risk preferences of the decision maker. A value of β close to 0 indicates that the decision maker is willing to accept a higher level of risk in order to achieve lower expected total costs. If the decision maker is considered to be rather risk averse, then β might be chosen closer to 1 to focus on price risk minimisation while tolerating possibly higher expected total costs.

4 Case Study and Computational Results

In this case study, the efficient performance of the proposed MILP model is illustrated using the example of German energy markets. The time horizon is set to 28 days. It is assumed that the industrial company has the option to sign a baseload contract with variable costs of 20.5 euro/MWh and an open contract with fixed costs of 100 euro and variable costs of 22.5 euro/MWh. The gas storage facility has a working gas volume of 500 m^3 and an injection/extraction rate of 50 m^3/h. Information on storage costs were obtained from the storage fee calculator of RWE AG (see http://www.RWE.com). The operating region of the CHP plant is specified as presented in [7] with a maximum heat and power output of 55 MW_{th} and 60 MW_{el}, respectively. Uncertain gas spot prices are described by a mean-reverting model, whereas a seasonal ARIMA model is used to capture short-term dynamics of electricity spot prices. The stochastic processes were fitted to historical data obtained from EEX and EPEX SPOT and used to generate price scenarios. Since the number of scenarios to be used in the optimisation model is limited, scenario reduction techniques are applied. We make use of the forward selection algorithm as proposed in [3].

To study the influence of different risk preferences on the optimal gas supply portfolio allocation, the proposed MILP model is exemplarily solved for the weighting factor β taking the values 0, 0.5 and 1. Figure 2 shows the average percentage shares of different gas procurement options of the total gas quantity procured over 100 scenarios for different values of β. First of all, gas storage capacity is not rented in any of the three cases. Apart from that, the optimal gas supply portfolio significantly differs for the different values of β. For example, if $\beta = 0$, then the baseload contract is not signed and almost three-quarters of the gas quantity is purchased from the spot market. By contrast, for $\beta = 1$ the share of the baseload contract accounts for 63% of the total gas quantity, whereas spot purchases only reach 25%.

The influence of different risk preferences is also reflected in the expected total costs and CVaR of the three alternatives. Corresponding results of the MILP model

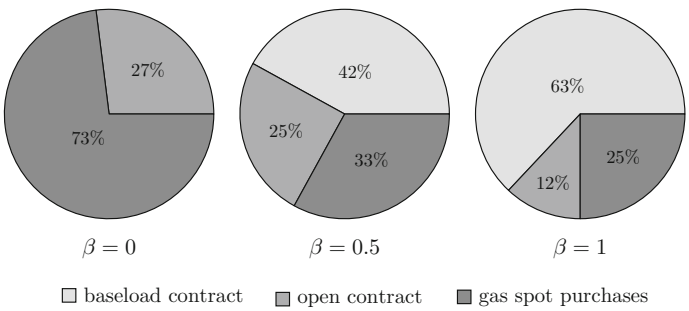


Fig. 2 Average percentage shares of the different gas procurement options of the total gas quantity procured over 100 scenarios for different values of the weighting factor β

Table 1 Results of the MILP model for different values of the weighting factor β

	Expected value (in mio. euro)	CVaR (in mio. euro)
$\beta = 0$	3.425	3.643
$\beta = 0.5$	3.431	3.621
$\beta = 1$	3.596	3.620

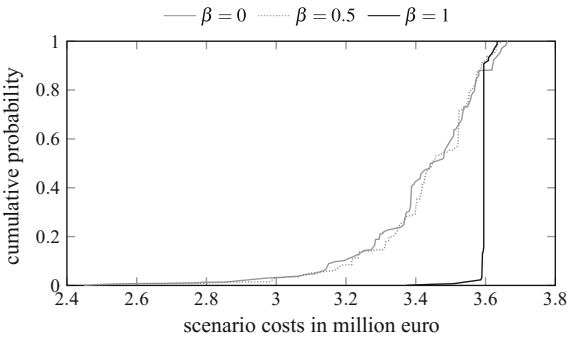


Fig. 3 Cum. distribution functions of the scenario costs for different values of the weighting factor β

are shown in Table 1. As one would expect, the lowest expected total costs are achieved for $\beta = 0$, whereas the lowest CVaR is given in the case of $\beta = 1$. However, it is worth pointing out that for $\beta = 1$ the CVaR only scarcely noticeable decreases, whereas expected total costs considerably increase by 4.8% compared to $\beta = 0.5$. An insight into the scenario costs is provided in Fig. 3 which displays the cumulative distribution functions for different values of β. It shows that even though the standard deviation is significantly lower for $\beta = 1$, only in very few cases total scenario costs are lower compared to the case of $\beta = 0$ and $\beta = 0.5$, respectively.

5 Conclusion

Operators of gas-fired power plants have to take gas procurement and power plant resource planning simultaneously into account. A two-stage MILP model was proposed to optimise the natural gas supply portfolio and CHP plant operation according to revenue potential in the electricity spot market. Price uncertainty was handled by means of stochastic processes. A convex combination of expected total costs and CVaR allows for representing different risk preferences of the decision maker. Results of a case study revealed the significant influence of the risk preference on the optimal gas supply portfolio as well as on expected total costs and CVaR.

References

1. Aouam, T., Rardin, R., Abrache, J.: Robust strategies for natural gas procurement. Eur. J. Oper. Res. **205**(1), 151–158 (2010)
2. Federal Statistical Office: Electricity generation plants of industry with an electric bottleneck capacity (gross) of 1 megawatt or more. Electricity and heat generation by energy source 2014 (2016)
3. Heitsch, H., Römisch, W.: Scenario tree modeling for multistage stochastic programs. Math. Program. **118**(2), 371–406 (2009)
4. Koberstein, A., Lucas, C., Wolf, C., König, D.: Modeling and optimizing risk in the strategic gas purchase planning problem of local distribution companies. J. Energ. Markets **4**(3), 47–68 (2011)
5. Lahdelma, R., Hakonen, H.: An efficient linear programming algorithm for combined heat and power production. Eur. J. Oper. Res. **148**(1), 141–151 (2003)
6. Rockafellar, R.T., Uryasev, S.: Conditional value-at-risk for general loss distributions. J. Bank. Finan. **26**(7), 1443–1471 (2002)
7. Vasebi, A., Fesanghary, M., Bathaee, S.M.T.: Combined heat and power economic dispatch by harmony search algorithm. Electr. Power Energ. Syst. **29**(10), 713–719 (2007)

Benders Decomposition on Large-Scale Unit Commitment Problems for Medium-Term Power Systems Simulation

Andrea Taverna

Abstract The Unit Commitment Problem (UCP) aims at finding the optimal commitment for a set of thermal power plants in a Power System (PS) according to some criterion. Our work stems from a collaboration with RSE S.p.A., a major industrial research centre for PSs in Italy. In this context the UCP is formulated as a large-scale MILP spanning countries over a year with hourly resolution to simulate the ideal behaviour of the system in different scenarios. Our goal is to refine existing heuristic solutions to increase simulation reliability. In our previous studies we devised a Column Generation algorithm (CG) which, however, shows numerical instability due to degeneracy in the master problem. Here we evaluate the application of Benders Decomposition (BD), which yields better conditioned subproblems. We also employ Magnanti-Wong cuts and a "two-phases scheme", which first quickly computes valid cuts by applying BD to the continuous relaxation of the problem and then restores integrality. Experimental results on weekly instances for the Italian system show the objective function to be flat. Even if such a feature worsens convergence, the algorithm is able to reach almost optimal solutions in few iterations.

1 Model

We used the model described in [2]. In the following, for brevity, we report only the most important elements used in the Benders Decomposition. In each zone thermal plants are divided in groups, characterised by the same marginal cost, and, inside each group, subgroups, characterised by the same technical minima, maxima and fixed cost term.

Sets

Let T be the set of time periods, Z the set of zones, $A \subset Z \times Z$ the set of links between zones and G_z the set of thermal power plant groups for $z \in Z$. For each $g \in G_z$ a set of subgroups M_{zg} is defined.

A. Taverna (✉)
Università Degli Studi di Milano, via Saldini 50, Milan, Italy
e-mail: andrea.taverna@unimi.it

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_25

Parameters

For $t \in T$, $z \in Z$, $g \in G_z$ and $m \in M_{zg}$ let c_{tzg} be the marginal production cost at time $t \in T$ (€/MWh), e_{tzgm} the fixed term of the cost function at time $t \in T$ (€/h), p_{zgm} and P_{zgm} the minimum and maximum power produced by plants in subgroup $m \in M_{zg}$ (MW), and T^{on}_{tzgm} and T^{off}_{tzgm} be the periods in which the plant has to maintain state if turned on and off respectively at time $t \in T$.

For $t \in T$ and $z \in Z$ let d_{zt} be the zonal hourly demand. For $t \in T$ and $(i, j) \in A$ let b_{ij} be the maximum capacity of the link. Finally, for $t \in T$ let $VOLL_t$ be the Value Of Lost Load (VOLL) (€/MWh), i.e. the cost of not satisfying one unit of demand at time t, such that $VOLL_t \gg \max_{z \in Z, g \in G_z}\{c_{tzg}\}$.

Variables

For each period $t \in T$, zone $z \in Z$ and group $g \in G_z$ let x_{tzg} be the production level (MWh), y_{tzgm} the number of active plants of subgroup $m \in M_{zg}$, up_{tzgm} and dn_{tzgm} be the number of plants of family $m \in M_{zg}$ switched on and off at time t respectively.

For each link $(i, j) \in A$ let w_{tij} be the energy flowing through link $(i, j) \in A$ time $t \in T$ (MWh).

Finally, for each period $t \in T$, zone $z \in Z$ let ENP_{tz} be the Energy Not Provided in zone z, i.e. the unsatisfied amount of demand (MWh), EIE_{tz} the Energy In Excess, i.e. the excess production, and x^h_{zt} the amount of power provided, through production, or absorbed, through pumping, by the hydroelectric power plants in zone z.

Here follows the UCP model.

$$\min \ \phi = \sum_{\substack{t \in T, z \in Z, \\ g \in G_z}} c_{tzg} x_{tzg} + \sum_{\substack{t \in T, z \in Z, \\ g \in G_z, m \in M_{zg}}} e_{tzgm} y_{tzgm} + \sum_{t \in T, z \in Z} \text{ENP}_{tz} VOLL_t \tag{1a}$$

$$\text{s.t.:} \sum_{m \in M_{zg}} p_{zgm} \cdot y_{tzgm} \le x_{tzg} \le \sum_{m \in M_{zg}} P_{zgm} \cdot y_{tzgm} \quad \forall t \in T, z \in Z, g \in G_z \tag{1b}$$

$$\text{up}_{tzgm} \ge y_{tzgm} - y_{(t-1)zgm} \quad \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{1c}$$

$$\text{dn}_{tzgm} \ge y_{(t-1)zgm} - y_{tzgm} \quad \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{1d}$$

$$y_{tzgm} \ge \sum_{\tau \in T : t \in T^{\text{on}}_{\tau}} \text{up}_{\tau zgm} \quad t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{1e}$$

$$y_{tzgm} \le |M_{zgm}| - \sum_{\tau \in T : t \in T^{\text{off}}_{\tau}} \text{dn}_{\tau zgm} \quad t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{1f}$$

$$x^h_{tz} + \sum_{g \in G_z} x_{tzg} + \sum_{(i,z) \in A} w_{tiz} + \text{ENP}_{tz} \ge d_{tz} \sum_{(z,j) \in A} w_{tzj} + \sum_{z \in Y} \text{EIE}_{tz} \quad \forall t \in T, z \in Z \tag{1g}$$

$$y_{tzgm}, \mathrm{up}_{tzgm}, \mathrm{dn}_{tzgm} \in [0, |M_{zgm}|] \cap Z_0^+ \quad \forall t \in T, z \in Z, g \in G_z, m \in M_{zg} \tag{1h}$$

$$x_{tz}^h \in H^{tz} \quad \forall t \in T, z \in Z \tag{1i}$$

$$w_{tij} \in [0, B_{ij}] \quad \forall t \in T, (i,j) \in A \tag{1j}$$

$$ENP_{tz} \geq 0, EIE_{tz} \geq 0 \quad \forall t \in T, z \in Z \tag{1k}$$

The objective (1a) is to minimise the production costs of thermal power plants and the costs of energy not provided. Constraints (1b) force the production of thermal plants to respect its technical limits when active. Constraints (1c)–(1f) are minimum up/down constraints as specified in [4]. Constraints (1g) force balance of network flows. Hydroelectric production $(x_{tz}^h)_{t\in T, z\in Z}$ in each period and zone is assumed to have negligible marginal production costs and to follow a linear network flow model corresponding to polytopes H^{tz}.

The UCP model Eq. (1) is a large-scale mixed-integer linear problem.

2 Benders Decomposition

We apply Benders Decomposition [1] to model (1) by dualising constraints (1b) which couple thermal production levels and the state of the plants, yielding a linear continuous dispatching model

$$\min \quad \phi(\tilde{\mathbf{y}}) = \sum_{\substack{t\in T, z\in Z,\\ g\in G_z}} c_{tzg} x_{tzg} + \sum_{\substack{t\in T, z\in Z,\\ g\in G_z, m\in M_{zg}}} e_{tzgm} \tilde{y}_{tzgm} + \sum_{t\in T, z\in Z} \mathrm{ENP}_{tz} VOLL_t \tag{2a}$$

$$\sum_{m\in M_{zg}} p_{zgm} \cdot \tilde{y}_{tzgm} \leq x_{tzg} \leq \sum_{m\in M_{zg}} P_{zgm} \cdot \tilde{y}_{tzgm} \quad \forall t \in T, z \in Z, g \in G_z \tag{2b}$$

$$(1g), (1i), (1k) \tag{2c}$$

which depends on a feasible commitment $\tilde{\mathbf{y}}$ for thermal plants, and a pure integer master problem

$$\min \quad \psi \tag{3a}$$

$$(1c) - (1f), (1h) \tag{3b}$$

$$\psi \geq \sum_{\substack{t\in T, z\in Z,\\ g\in G_z, m\in M_{zg}}} (e_{tzgm} + \mu_{tzgb} p_{zgm} - \lambda_{tzgb} P_{zgm}) y_{tzgm} + \eta_b \quad \forall b \in B \tag{3c}$$

which computes new commitments for thermal units. Equation (3c) is Benders optimality cuts with μ_{tzgb} and λ_{tzgb} being the dual variables associated to the lower and upper bounds in Eq. (1b) respectively and η_b is the constant term derived from the objective value of model (2).

To further improve the efficacy of the method, we employed Magnanti-Wong cuts (MW) (see [6] for a recent study). Given a dual feasible solution of Eq. (2) the MW method solves a linear problem to find a new dual solution which (i) is feasible for Eq. (2), (ii) lies on its optimal facet and (iii) is closer to an interior point of model Eq. (3). MW cuts are guaranteed to be Pareto-optimal, in the sense that they yield the tightest bound among the Benders optimality cuts for the current master solution.

In practice constraint (ii) can be numerically unstable and lead to numerical unboundness [6]. In our scheme, when this issue is encountered, the algorithm just adds the original Benders cuts.

3 Experiments and Conclusions

We conducted a series of experiments on ten "weekly" instances of 168 h obtained from a scenario hypothesis from RSE S.p.A. for Italy in 2011. For details on the instances we refer to [2]. The experiments were implemented with AMPL 20081120 and CPLEX 12.6 on a Linux laptop with 4 GB RAM and 2.7 GHz quad-core processor. For comparison the instances require around 10 min to be solved by the CPLEX MIP solver on the same system.

The method initialises the BD algorithm with a heuristic solution obtained from a variant of the algorithm presented in [2] (see [3] for more details). Then the BD procedure performs a two-phase algorithm [5] starting with k iterations of the first phase, where integrality constraints on the master problem are relaxed, and c iterations on the second phase, where the integrality constraints are re-introduced. We evaluated the algorithm with and without applying the MW cuts at each iteration. The procedure is then described by a triple of parameters (k, c, m) where $m \in \{N, Y\}$ and $m = Y \iff$ MW cuts are computed. We considered 6 possible combinations of parameters values, yielding 60 different tests.

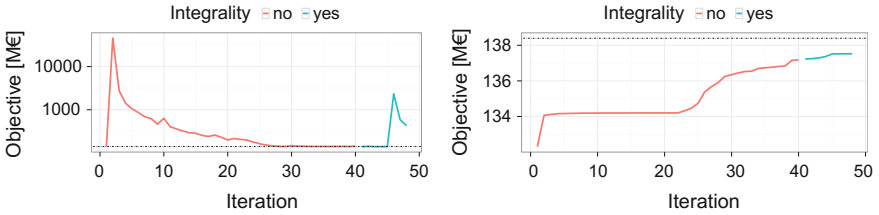
In Table 1 we report the initial gap obtained from the heuristic and its computation time for the ten instances. In Table 2 we report for each configuration, across instances, the following indicators: average total computing time, average number of failed MW cut computations (due to numerical unboundness), number of instances for which the upper bound improved, average gap, in percentage, between the best primal and dual bounds for each instance, which estimates the distance of the algo-

Table 1 Initial heuristic: gap and computation time

Instance	Week-5	Week-10	Week-15	Week-20	Week-24	Week-30	Week-35	Week-40	Week-45	Week-50
Gap %	1.28	1.42	1.56	2.23	1.03	0.93	0.83	1.27	1.54	1.3
Time (s)	11	11	11	10	11	11	11	11	11	12

Table 2 Results for different configurations across instances

1st phase	2nd phase	MW cuts	Avg. Time (s)	Avg. MW failures	# UB improvements	Avg. BD Gap %	Avg. UB improvement %	Avg. LB improvement %
20	8	N	959	–	1	27.4	−23.3	−2.1
20	8	Y	1460	4.5	1	30.5	−26.2	−2.2
20	16	N	1994	–	2	12.8	−9.5	−1.7
20	16	Y	2787	9.2	4	11.6	−8.4	−1.6
40	8	N	2344	–	3	12.1	−9.6	−1.0
40	8	Y	3264	18	5	4.7	−2.2	−1.0



(a) Slave (upper bound) (b) Master (lower bound)

Fig. 1 Bounds for the BD algorithm on week-10 with configuration $(k, c, m) = (40, 8, Y)$. *Dashed lines* represent the initial upper and lower bound respectively

rithm from termination, and average improvement over the upper bound and the lower bound, in percentage term, compared to the initial values from the heuristic.

For illustration purposes, in Fig. 1 we report the series of primal and dual bounds during the BD algorithm on the week-10 instance with configuration $(k, c, m) = (40, 8, Y)$.

Figure 1 suggests the algorithm encounters a "plateau" after the first iterations, where the dual bound "stalls", i.e. remains almost constant. Once the plateau is overcome, both bounds start improving again. This agrees with our previous experience with CG [3], where we encountered numerical issues due to degeneracy in the master problem.

Results in Table 2 show the most effective configurations for the BD algorithm are those employing more iterations for the first phase. The result can be explained considering that for this type of UCP model the integrality gap can be quite small, especially for instances with longer horizons. Hence optimality cuts for the continuous relaxation can be quite effective for the original problem. The presence of degenerate solutions and flatness in the objective function imply the dual information obtained from decoupling commitments of power plants and their dispatching, either through Benders Decomposition or Column Generation, can be inadequate to efficiently determine solutions with increasing accuracy. On the other hand, Table 2 shows MW cuts, by exploiting geometric characteristics of the problem's polytope,

can be quite effective, especially once the initial plateau is overcome, to improve the algorithm's convergence, despite the fact that the separating problem can fail in 20–50% of the iterations.

We investigated the use of Benders Decomposition to solve large-scale medium-term UCPs. Previous studies with Column Generation showed degeneracy and objective flatness could cause numerical instability in state-of-the-art LP solvers on larger instances. In this work we verified the same issues can cause stalling and slow convergence using Benders Decomposition, as cuts delineate nearly flat regions of the objective function. On the other hand, by exploiting information embedded in the continuous relaxation of the model and geometric properties of Magnanti-Wong cuts, significant improvements in effectiveness of the algorithms can be readily obtained.

References

1. Benders, J.: Partitioning procedures for solving mixed-variables programming problems. Comput. Manag. Sci. **2**(1), 3–19 (1962)
2. Ceselli, A., Gelmini, A., Righini, G., Taverna, A.: Mathematical programming bounds for large-scale unit commitment problems in medium-term energy system simulations. In: SCOR 2014—4th Student Conference on Operational Research (2014)
3. Ceselli, A., Righini, G., Siface, D., Taverna, A.: Large-scale optimization of the unit commitment problem for medium-term simulations of energy systems. In: Presented at Column Generation. https://www.gerad.ca/colloques/ColumnGeneration2016/index.html (2016)
4. Deepak, R., Takriti, S.: Minimum up/down polytopes of the unit commitment problem with start-up costs. IBM Research Report (2005)
5. Mercier, A., Cordeau, J.-F., Soumis, F.: A computational study of Benders decomposition for the integrated aircraft routing and crew scheduling problem. Comput. Oper. Res. **32**(6), 14511476 (2005)
6. Papadakos, N.: Practical enhancements to the magnantiwong method. Oper. Res. Lett. **36**(4), 444–449 (2008)

Deployment and Relocation of Semi-mobile Facilities in a Thermal Power Plant Supply Chain

Tobias Zimmer, Patrick Breun and Frank Schultmann

Abstract Co-firing of biomass in coal-fired power plants is considered one of the most economic ways of carbon dioxide abatement. We investigate the deployment and relocation of several semi-mobile processing facilities in order to supply a large coal-fired power plant with high-quality renewable energy carriers. Semi-mobile facilities are characterized by a containerized design and can be relocated in case of changes in supply and demand. The energy carriers which are produced by different types of semi-mobile technologies are bulky goods with high density and properties comparable to those of coal and fuel oil. Thus, intermodal transportation is required to achieve transportation costs which are competitive with the delivered cost of fossil fuels at the plant's gate. The optimization of the investigated supply chain therefore requires simultaneous planning of semi-mobile facility deployment and intermodal transportation. To this end, we present a mixed-integer linear problem which optimizes the number of semi-mobile facilities, their respective relocation over time and the intermodal transportation of produced energy carriers to the power plant. In the presented case, train transportation is characterized by a low geographical coverage of the railway network and restrictions representing minimum shipping volumes per railway line. The model minimizes the objective function of total supply chain costs including electricity generation, transportation, the operation and relocation of the semi-mobile plants and the necessary forestry operations associated with the deployed facilities. The model is implemented in GAMS and solved using the CPLEX solver. We discuss a numerical example based on data from the forestry and energy sector in Chile.

1 Introduction

Co-firing of biomass with coal has been identified as a cost-efficient way to reduce the carbon dioxide emissions of electricity production. However, coal-fired power plants frequently reject biomass as a substitute fuel due to its insufficient

T. Zimmer (✉) · P. Breun · F. Schultmann
Karlsruhe Institute of Technology (KIT), Institute for Industrial Production,
Hertzstr. 16, 76187 Karlsruhe, Germany
e-mail: tobias.zimmer@kit.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_26

combustion properties and high costs of transport and logistics. Pre-treatment processes such as pelletization and torrefaction produce densified bioenergy carriers which can be transported efficiently and can be easily grinded and mixed with coal. Optimization problems for co-firing and pre-treatment supply chains have been presented in [4] and in [3]. While the investigated supply chains enable long-distance transportation to a power plant, high costs are still associated with the collection of feedstock which is highly dispersed along the territory. It has therefore been suggested to deploy mobile facilities to carry out pre-treatment processes directly at the harvesting site [1]. Optimization problems for supply chains with mobile facilities were studied in [2, 5]. Both articles investigate mobile facilities from a strategic perspective and present a supply chain model based on a facility location problem. In this paper, we aim to explore the deployment of mobile facilities from a tactical perspective. To this end, we introduce a problem which includes the scheduling of facility relocations between harvesting sites.

2 The Semi-mobile Facility Relocation Problem

The deployment and relocation problem is modelled as a mixed integer linear program (MILP). Let L be a set of locations and T be the number of periods in the planning horizon. The feedstock supply at location $i \in L$ in period $t \in T$ is represented by S_{it}. Bioenergy carriers are produced from biomass feedstock at a constant production rate R. The maximum operating days of a facility per year are given by D. The time τ is required to relocate a semi-mobile facility from one location to another. In comparison to a mobile plant directly mounted on a single truck, we define a semi-mobile plant as a larger facility which requires approximately five trucks for relocation. Changing the location takes approximately two weeks, therefore a facility should be operating at a location for several weeks or months. It is assumed that the time required for a relocation is determined mainly by the disassembly and assembly of the facility while the distance-dependent relocation time is negligible. The operating cost of a facility and the relocation cost are given by C^{OP} and C^{RE}, the feedstock cost is denoted by C^{FS}. Bioenergy carriers produced by semi-mobile facilities are transported to transshipment terminals by truck. A subset $G \subset L$ of all locations are connected to the railway network and can serve as transshipment terminals. Transportation distances between harvesting sites and transshipment terminals are denoted by D_{ig}. The energy carriers are subsequently transported by train to coal-fired power plants represented by the set $H \subset L$. Co-firing at each power plant h is limited to a maximum amount F_h of bioenergy carriers. Transportation distances on the railway network are given by D'_{gh}. Due to economies of scale, train transportation is only available if the minimum shipping volume χ_{gh} of a railway line is fulfilled. Specific costs of truck and train transportation are denoted by C^{TG} and C^{TH}. M represents a sufficiently large number used to model binary conditions. The objective function (1) corresponds to the savings which can be achieved by replacing coal with biomass.

The price of coal is given by P and includes the market price of coal plus a carbon dioxide tax. The decision variables are given below:

α_{it}	starting time of operations at location $i \in L$ in period $t \in T$ (in days)
ω_{it}	end time of operations at location $i \in L$ in period $t \in T$ (in days)
k	integer variable representing the number of semi-mobile facilities
z_{ijt}	binary variable indicating a relocation from $i \in L$ to $j \in L$ in period t
y_{it}^{ini}	binary variable indicating if $i \in L$ is the initial position of a facility in t
y_{it}^{end}	binary variable indicating if $i \in L$ is the end position of a facility in t
y_{it}^{av}	binary variable indicating if a facility is available at $i \in L$ in t
a_{it}	amount of bioenergy carriers produced at location i in period t (in tons)
x_{igt}	amount of products transported from facilities to terminals (in tons)
x'_{ght}	amount of products transported from terminals to power plants (in tons)
γ_{ght}	binary variable indicating transportation from terminal g to power plant h

Based on these definitions, the model can be specified as follows:

$$\max \sum_{g\in G}\sum_{h\in H}\sum_{t\in T} P\, x'_{ght} - \sum_{i\in L}\sum_{t\in T} C^{FS} a_{it} - \sum_{i\in L}\sum_{g\in G}\sum_{t\in T} C^{TG} D_{ig} x_{igt} - \sum_{g\in G}\sum_{h\in H}\sum_{t\in T} C^{TH} D'_{gh} x'_{ght} - C^{OP} k\, D - \sum_{i\in L}\sum_{j\in L}\sum_{t\in T} C^{RE} z_{ijt} \tag{1}$$

subject to:

$$a_{jt} \le R(\omega_{jt} - \alpha_{jt} - \tau \sum_{i\in L} z_{ijt}) \quad \forall j \in L, \forall t \in T \tag{2}$$

$$a_{it} \le S_{it} \quad \forall i \in L, \forall t \in T \tag{3}$$

$$\sum_{i\in L}(\omega_{it} - \alpha_{it}) = kD \quad \forall t \in T \tag{4}$$

$$\alpha_{it} \le \omega_{it} \le D \quad \forall i \in L, \forall t \in T \tag{5}$$

$$\alpha_{jt} - \omega_{it} - M(1 - z_{ijt}) \le 0 \quad \forall i, j \in L, \forall t \in T \tag{6}$$

$$\omega_{it} - \alpha_{it} \le M y_{it}^{av} \quad \forall i \in L, \forall t \in T \tag{7}$$

$$\sum_{j\in L} z_{ijt} + y_{it}^{end} = y_{it}^{av} \quad \forall i \in L, \forall t \in T \tag{8}$$

$$y_{jt}^{ini} + \sum_{i\in L} z_{ijt} = y_{jt}^{av} \quad \forall j \in L, \forall t \in T \tag{9}$$

$$\alpha_{it} - M(1 - y_{it}^{ini}) \le 0 \quad \forall i \in L, \forall t \in T \tag{10}$$

$$D\, y_{it}^{end} \le \omega_{it} \quad \forall i \in L, \forall t \in T \tag{11}$$

$$\sum_{i\in L} y_{it}^{ini} = \sum_{i\in L} y_{it}^{end} = k \quad \forall t \in T \tag{12}$$

$$y_{it}^{ini} = y_{i,t-1}^{end} \quad \forall i \in L, t = 2, ..., T \tag{13}$$

$$\sum_{g \in G} x_{igt} \leq a_{it} \quad \forall i \in L, \forall t \in T \tag{14}$$

$$\sum_{i \in L} x_{igt} = \sum_{h \in H} x'_{ght} \quad \forall g \in G, \forall t \in T \tag{15}$$

$$\sum_{g \in G} x_{ght} \leq F_h \quad \forall h \in H, \forall t \in T \tag{16}$$

$$x'_{ght} - M\gamma_{ght} \leq 0 \quad \forall g \in G, \forall h \in H, \forall t \in T \tag{17}$$

$$\gamma_{ght}\chi_{gh} \leq x'_{ght} \quad \forall g \in G, \forall h \in H, \forall t \in T \tag{18}$$

Constraint (2) ensures that the production of bioenergy carriers does not exceed the time a semi-mobile facility is available at the respective location. The production also cannot exceed the feedstock supply (3). The total operation time at all locations is limited by the number and the capacity of the deployed facilities (4). The end time of each operation is limited by the length of a period and must be greater than the starting time (5). A facility can only be relocated if the end time of the departed location corresponds to the starting time of the receiving location (6). Operations can only take place if a facility is available at a location during the respective period (7). If a facility is available at location i in period t, it has either been started there or has been relocated from another location (8). Likewise, if a facility is available at location i in period t, it can either stay there until the end of the period or move to another location (9). Any initial position corresponds to a starting time of zero (10) while any end position corresponds to an end time of D (11). The number of initial positions and end positions must be equal to the number of facilities (12). If a location is the end position of a facility in period t, it is also the initial position of a facility in period $t + 1$ (13). The amount of products transported to transshipment terminals is limited by the production (14) and equals the quantity transported to the power plants (15). Each plant has a maximum amount of biomass that can be co-fired with coal (16). Constraints (17) and (18) guarantee that transshipment is only possible if the minimum shipping volume is fulfilled.

3 Reduction of the Number of Input Locations

The presented problem involves several integer variables to model the relocation of the semi-mobile facilities. Solving instances with high spatial granularity and a high number of locations included in L can therefore lead to long computation times. However, the structure of the railway network and the locations of the power plants

allow to construct a heuristic which reduces the set of locations L to a smaller set V consisting of promising locations:

- $V = \emptyset$. For all $t \in T$ compute V_t:
  - Compute $g_i = \arg\min(D_{ig})$ and $h_i = \arg\max(D'_{g_i,h})$ for all $i \in L$.
  - Compute $\widetilde{CT}_{it} = C^{TG} D_{i,g_i} + C^{TH} D'_{g_i,h_i}$ for all $i \in L$.
  - Compute $\widetilde{C}_{it} = \widetilde{CT}_{it} + \frac{C^{OP}(\frac{S_{it}}{R}+\tau)+C^{RE}}{S_{it}}$ for all $i \in L$.
  - Set $U_t = L$. While $\sum_{i \in V_t} S_{it} \leq \sum_{h \in H} F_h$ add the element $i \in U_t$ with the minimum $\widetilde{C}_{it}$ to V_t and remove it from U_t.
  - Add all elements $i \in U_t$ with $\widetilde{CT}_{it} \leq \max_{i \in V_t}(\widetilde{CT}_{it})$ to V_t.
- $V = V_1 \cup V_2 \cup \cdots \cup V_T$.

4 Computational Results

The model is applied to a case study in the south of Chile. The included regions consist of 195 municipalities which are accepted as the locations L of the model. The planning period is 4 years. The feedstock supply corresponds to approximately 600,000 tons of forest residues. The supply at each location changes annually according to the respective forest management plan. Bioenergy carriers are delivered to a single power plant located in the city of Concepción. The production rate and the relocation time of a semi-mobile facility are set to 100 t/day and 14 days. Truck and train transportation costs are set to 0.2 EUR/t-km and 0.05 EUR/t-km. The model was implemented in GAMS and solved using IBM ILOG CPLEX on an Intel Core i5-5200U CPU with 2.2 GHz and 12 GB RAM. Computational results for the presented model and the proposed heuristic are given in Table 1. The optimum schedule of facility relocations and transports in a single period is illustrated by Fig. 1.

Table 1 Computational results

Instance		Full set of locations				Reduced set of locations		
T	F	Facilities	ζ[(a)]	Objective[(b)]	Time (s)	$\lvert V \rvert$	Gap[(c)] (%)	Time (s)
1	60 kton	2	8	337	64	64	0.43	8
2	60 kton	2	16	661	1,171	70	1.75	400
3	60 kton	2	24	975	11,982	72	0.04	750
4	60 kton	2	32	1,331	31,075	80	2.23	2,012
3	80 kton	3	45	1,271	7,807	64	0.73	701
3	100 kton	3	48	1,488	17,609	67	0.36	827

(a) ζ: number of relocations, (b) in 1000 EUR, (c) to optimum solution with set L

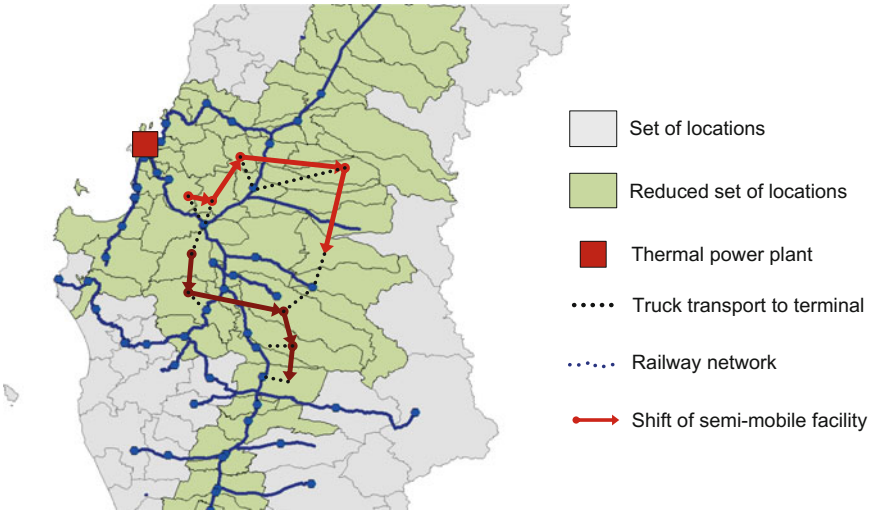


Fig. 1 Facility relocations on the reduced set of locations for $T = 1$

5 Conclusion

We presented a model for the deployment and relocation of semi-mobile facilities in a power plant supply chain. The model determines the optimum schedule of the facilities while taking into account the structure of the transportation network. Solving instances with a high number of locations can lead to long computation times. It was shown that by using a heuristic to reduce the number of locations, the problem can be solved within a reasonable computation time. The results of a case study in southern Chile indicate that semi-mobile facilities enable an efficient utilization of fluctuating biomass potentials.

References

1. Brown, D., Rowe, A., Wild, P.: A techno-economic analysis of using mobile distributed pyrolysis facilities to deliver a forest residue resource. Bioresour. Technol. **150**, 367–376 (2013)
2. Mirkouei, A., Mirzaie, P., Haapala, K.R., Sessions, J., Murthy, G.S.: Reducing the cost and environmental impact of integrated fixed and mobile bio-oil refinery supply chains. J. Clean. Prod. **113**, 495–507 (2016)
3. Pérez-Fortes, M., Laínez-Aguirre, J.M., Bojarski, A.D., Puigjaner, L.: Optimization of pretreatment selection for the use of woody waste in co-combustion plants. Chem. Eng. Res. Des. **8**, 1539–1562 (2014)
4. Roni, Md.S., Eksioglu, S.D., Searcy, E., Jha, K.: A supply chain network design model for biomass co-firing in coal-fired power plants. Transp. Res. Part E: Logist. Transp. Rev. **61**, 115–134 (2014)
5. Sharifzadeh, M., Garcia, M.C., Shah, N.: Systematic decision-making for centralized, distributed, and mobile biofuel production using mixed integer linear programming (MILP) under uncertainty. Biomass Bioenergy **81**, 401–414 (2015)

Part VII
Finance

Applying a Novel Investment Evaluation Method with Focus on Risk—A Wind Energy Case Study

Jan-Hendrik Piel, Felix J. Humpert and Michael H. Breitner

Abstract Renewable energy investments are typically evaluated using traditional discounted cash flow (DCF) methods, such as the net present value (NPV) or the internal rate of return (IRR). These methods utilize the discount rate as an aggregate proxy for risk and the time value of money, which leads to an inadequate modeling of risk. An alternative to these methods represents the decoupled net present value (DNPV). Instead of accounting for risk in the discount rate, the DNPV utilizes so-called synthetic insurance premiums. These allow for the individual and disaggregate pricing of risk and can enhance the quality of investment decisions by facilitating a more detailed and comprehensive representation of the underlying risk structure. To reliably estimate and forecast synthetic insurance premiums requires the availability of appropriate data and expertise in interpreting this data. Thus, the practicality of the results calculated based on the DNPV depends on the quality of the inputs and the expertise of the analyst. After reviewing the main theory of the DNPV, we apply the method to a wind energy investment case to demonstrate its applicability and prospects. To illustrate the calculation of the synthetic insurance premiums, selected risk factors are modeled with probability distributions via Monte Carlo simulation (MCS). Our results show that the DNPV's seamless integration of risk assessment with investment evaluation is a promising combination and warrants further research.

J.-H. Piel (✉) · F.J. Humpert · M.H. Breitner
Information Systems Institute, Leibniz University Hannover,
Königsworther Platz 1, 30167 Hannover, Germany
e-mail: piel@iwi.uni-hannover.de

F.J. Humpert
e-mail: humpert@iwi.uni-hannover.de

M.H. Breitner
e-mail: breitner@iwi.uni-hannover.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_27

1 Introduction

In theory [1] and practice [2–5], DCF methods, such as the NPV and the IRR, are often used for evaluating investments in infrastructure and renewable energy projects. Despite their popularity, their weaknesses and limitations are widely recognized in the scientific literature [1, 3, 6]. Most critical, but difficult is the selection of an appropriate discount rate in DCF analysis [2]. Often used are risk-adjusted discount rates (RADRs). By adding risk-free rate and risk premium, RADRs aggregate the time value of money and risk in a single metric [2, 6].

However, the bundling of time preference and risk in the discount rate obscures the appropriate modeling of investment risks. For instance, in the case of negative cash flows, selecting a higher RADR to account for an increase in risk, produces a more favorable NPV. Using RADRs is therefore a rather inconsistent way to account for risk [2, 7]. Consequently, the use of DCF methods based on RADRs distorts investment evaluations and can result in misguided investment decisions [8]. Even supplementing these methods with more sophisticated approaches, such as using probability distributions in combination with MCS and real option valuation [7] cannot overcome the problem of discount rate selection. This is for instance discussed by [9] with respect to the use of MCS in investment evaluations.

A solution to the shortcomings described above is the DNPV, which was first introduced by [2, 7]. It solves the issues surrounding discount rate selection by decoupling risk from the time value of money [2]. Further, it allows to deal with systematic and unsystematic risks individually [7]. Both is achieved through so-called synthetic insurance premiums (SIPs). Investors, as equity providers, are the last to be paid from investments' returns and absorb the losses when risks materialize. Consequently, the DNPV treats investors as insurance providers for any risk not allocated to third parties through risk management measures [8]. This being the case, SIPs are priced risks that have to be treated as costs to an investment. They render an investment's cash flows riskless and thereby legitimize discounting at the risk-free rate [2]. In addition, SIPs can help assess and communicate the degree to which an investment is expected to reward investors for taking on risk [2]. As a result, the DNPV can support a more thorough analysis of the risk profile of investments and can provide a broader and more consistent foundation for investment decisions.

Wind energy projects are technically complex, highly leveraged, illiquid and capital intensive investments. Comprehensively analyzing the risks of such investments and their impact on profitability is of particular importance, as these characteristics potentially heighten the exposure to unsystematic risks for a given investor. By applying the DNPV to a solar energy project, [8] were the first to demonstrate the DNPV's feasibility in the context of renewable energy investments. An application of the method to a wind energy case is still missing from the literature. We aim to address this gap by providing methodological support tailored to the needs of wind energy investors. We implement the DNPV and its related concepts in MATLAB and utilize probability distributions generated via MCS for modeling risk. In order to demonstrate the DNPV's prospects and functionality, we illustrate its application with a stylized wind energy investment case.

2 Wind Energy Investment Case

The design of our investment case is based on recent data from the German wind energy market [10] as well as a risk breakdown structure template for renewable energy projects by [11]. Within the case the perspective of a consortium of investors in negotiations with a project developer over a 70% stake in a fully developed and operational wind energy project is adopted. The investors want to negotiate a reasonable price for the investment such that they can expect to be compensated with a return for taking on the risks of the project. To support their negotiations, the DNPV in combination with the NPV is applied. The remaining operating life of the project is 19.5 years, whereas an additional decommissioning of six months is expected. Table 1 presents the expected revenues and operating expenditures (OPEX).

The project is organized as a special purpose vehicle with a debt ratio of 85%. The debt is provided in the form of an annuity loan of €15,903 T with an interest rate of 2.5%. Its repayment starts at the beginning of the third year of operation. OPEX increase with an inflation rate of 1%. In previous auctions, the project has been awarded a feed-in tariff of €85/MWh. The wind park consists of four turbines with an installed capacity of 2.5 MW each. The expected full load hours before losses amount to 2,933.55 h. The total park losses of 11.49% are a function of various influencing factors, such as wake losses and turbine availability. Consequently,

Table 1 Free cash flows to equity analysis of the investment case in thousand Euro (€T)

Parameter	Year 1	Year 2	Year 3	...	Year 20	Distribution[a]
Maintenance and repair	259.89	263.58	266.21	...	157.25	T(251.51, 90%, 120%)
Land lease	124.28	125.52	126.77	...	74.84	N(124.28, 10%)
Direct marketing	53.16	53.70	54.24	...	32.03	U(40.19, 66.13)
Other OPEX	194.06	195.98	197.96	...	116.94	N(194.06, 10%)
Total OPEX	**631.39**	**638.78**	**645.18**	**...**	**381.06**	
Decommissioning	**0.00**	**0.00**	**0.00**	**...**	**778.85**	U(311.40, 1,246.30)
Revenues before losses	2,493.52	2,491.30	2,491.60	...	1,245.45	N(2,493.52, 10%)
Losses monetarily	286.20	286.18	286.26	...	143.05	N(286.20, 15%)
Total revenues	**2,207.32**	**2,205.12**	**2,205.34**	**...**	**1,102.40**	
Corporate tax	122.08	73.46	30.96	...	211.26	
Debt service	0.00	1,162.60	1,954.10	...	0.00	
FCFE	**1,453.85**	**330.28**	**−424.90**	**...**	**−268.77**	

[a]Normal N(μ, σ in %); triangular T(mode, min in %, max in %); uniform U(min, max)

the expected annual electricity production equals to 25,964.85 MWh. This results in annual revenues of €2,207.32 T. The wind park is depreciated linearly over a period of 16 years and profits are subject to a corporate tax rate of 30%. The project's expected periodical free cash flows to equity (FCFE) are shown in Table 1. Individual risks in the case study are modeled using probability distributions in combination with MCS and 50,000 iterations as outlined in Table 1. The distribution types and shapes were selected based on recommendations by [11].

3 DNPV Analysis

Equation 1 outlines the concept for calculating the DNPV [2, 7] with V representing revenues, I expenditures and R SIPs. In line with [2], we understand SIPs as the fair insurance premiums, which compensate for expected losses resulting from unfavorable deviations of revenues and expenditures with respect to their expected values. In the numerator, for each period t, the respective SIPs reduce the expected revenues and increase the expected expenditures. To account for the time value of money, the resulting risk-adjusted cash flows are discounted at the risk-free rate r_f. This is legitimate given that the SIPs render the associated cash flows riskless [7].

$$DNPV = \sum_{t} \sum_{i,j} \frac{(\tilde{V}_{t,i} - \tilde{R}_{t,i}) - (\tilde{I}_{t,j} + \tilde{R}_{t,j})}{(1 + r_f)^t} \tag{1}$$

For the computation of SIPs, [2] distinguish between heuristic methods, stochastic processes, and the use of time-invariant probability distributions. Henceforth, we focus on the latter. When calculating SIPs based on probability distributions, differentiation between SIPs for expenditures and revenues is required. Equation 2 is to be used in the case of revenue risks [2, 7] where $\tilde{V}_{t,i}$ represents the expected revenues, $L_{t,i}$ the expected revenue shortfall relative to $\tilde{V}_{t,i}$ and $Pr\left[\tilde{V}_{t,i} > V_{t,i}\right]$ the probability of revenues falling below their expected value. To calculate SIPs for expenditure risks, Eq. 3 is to be used analogously [2, 7], with $\tilde{I}_{t,j}$ representing the expected expenditures, $L_{t,j}$ the expected excess expenditures relative to $\tilde{I}_{t,j}$ and $Pr\left[I_{t,j} > \tilde{I}_{t,j}\right]$ the probability of incurring excess expenditures.

$$\tilde{R}_{t,i} = (\tilde{V}_{t,i} - \tilde{V}^{-}_{t,i}) \cdot Pr\left[\tilde{V}_{t,i} > V_{t,i}\right] = L_{t,i} \cdot Pr\left[\tilde{V}_{t,i} > V_{t,i}\right] \tag{2}$$

$$\tilde{R}_{t,j} = (\tilde{I}^{+}_{t,j} - \tilde{I}_{t,j}) \cdot Pr\left[I_{t,j} > \tilde{I}_{t,j}\right] = L_{t,j} \cdot Pr\left[I_{t,j} > \tilde{I}_{t,j}\right] \tag{3}$$

To illustrate the calculation of SIPs, Fig. 1 shows the complete and truncated distributions for maintenance and repair (MR) and revenues before losses (RBL), including the characteristic inputs for the calculation of the corresponding SIPs.

Table 2 displays a breakdown of the total cost of risk represented by the SIPs for the parameters subject to risk. It gives an idea of how the DNPV integrates with risk

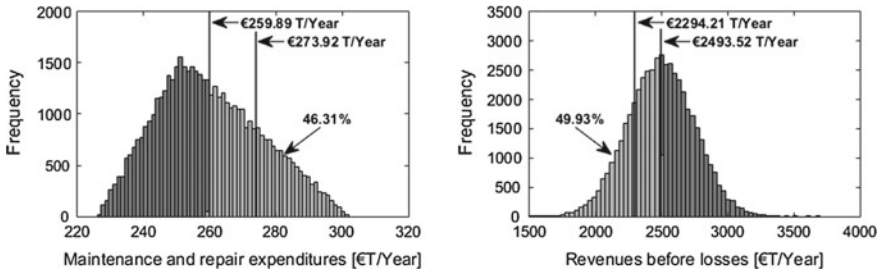


Fig. 1 SIP calculation for MR and RBL in year one. Applying Eq. 3 to the MR distributions results in an SIP of €6.50 T = (€273.92 T – €259.89 T) · 46.31%, whereas applying Eq. 2 to the RBL distribution gives an SIP of €99.52 T = (€2,493.52 T – €2,294.21 T) · 49.93%

Table 2 Decoupled FCFE and cost of risk described by SIPs in €T

Parameter	Year 1	Year 2	Year 3	...	Year 20	PV
Maintenance and repair	6.49	6.56	6.62	...	3.91	126.07
Land lease	4.97	5.02	5.07	...	2.99	96.38
Direct marketing	3.32	3.36	3.39	...	2.00	64.50
Other OPEX	7.74	7.82	7.90	...	4.67	150.26
Total OPEX	**22.52**	**22.76**	**22.98**	**...**	**13.57**	**437.21**
Decommissioning	**0.00**	**0.00**	**0.00**	**...**	**129.97**	**106.82**
Revenues before losses	99.52	99.34	99.27	...	49.71	1,762.30
Losses monetarily	16.18	16.18	16.16	...	8.09	286.83
Total revenues	**115.70**	**115.52**	**115.43**	**...**	**57.80**	**2,049.13**
Total SIPs	**138.22**	**138.28**	**138.41**	**...**	**201.34**	**2,593.16**
FCFE	**1,453.85**	**330.28**	**−424.90**	**...**	**−268.77**	**1,727.40 (= NPV)**
Decoupled FCFE	**1,315.63**	**192.00**	**−563.31**	**...**	**−470.11**	**3,363.50 (= DNPV)**

management by being able to quantify risks individually. For instance, total revenue risk results from adding the SIPs for RBL and the losses associated with the annual energy production, both expressed in monetary terms. RBL are the theoretical energy production if no park losses were to occur. The risk associated with RBL pertains to resource risk as well as the risk of inaccuracies in the wind data and modeling of the wind resource. Table 2 shows that revenue risk is the dominating risk category for the investment representing 79% of the total SIPs' present value (PV). OPEX risk is the second most important risk, but only a fraction of revenue risk with 16.9% of the total SIPs' PV. Although decommissioning risk outstrips OPEX risk in the final period, it is almost insignificant with 4.1%.

Deducting the SIPs from the FCFE yields the decoupled FCFE. Discounting these at the risk-free rate of 1% returns a DNPV of €3,363.50 T. To get the NPV of

€1,727.40 T, the FCFE are discounted at 8%, which is the required return assumed for the investors. Although the FCFE are more favorable than the decoupled FCFE, the DNPV exceeds the NPV, as the decoupled FCFE already price in risk. Due to this, the effect of discounting the FCFE in the NPV approach is significantly higher than the effect of discounting the decoupled FCFE in the DNPV approach.

The investors in the case study are well advised to proceed with the investment as long as they pay less than 70% of the NPV for the stake under negotiation. In this case, they are expected to earn a premium under the NPV and the DNPV paradigm. Thus, risk and the time value of money are expected to be covered according to both valuation approaches. However, based on the DNPV, investing even at a price higher than 70% of the NPV can be considered reasonable, whereas 70% of the DNPV can be considered the upper bound for a fair price. Exceeding this value means no longer being fairly compensated for the time value of money and potential losses associated with the investment. This case study demonstrates how the DNPV provides a new perspective on investment decisions by framing and modeling individual risks as costs to an investment. This facilitates a more thorough analysis of the risk structure of investments as well as their risk-return profile. Thus, the DNPV can broaden the foundation for investment decisions and thereby enhance their quality.

4 Limitations and Outlook

Decisions about wind energy investments require an adequate understanding of their risk-return profile. We applied the DNPV to the presented wind energy project to demonstrate its applicability. SIPs allow for a decoupling of the time value of money and risk and facilitate the pricing of risk. Further research has to explore how to assure the accuracy of SIPs, as project valuations require forecasting SIPs years into the future. In this respect, wind energy projects are ideal, since they are built in series, which facilitates data collection. Assets not sharing these characteristics appear to be less suitable for applying the DNPV. Espinoza [7] proposed the use of stochastic processes to calculate SIPs. Yet, the modeling of risks that are expected to behave dynamically still requires further research. In this regard the DNPV may profit from the experience with stochastic processes in the context of commodity price models used to evaluate long-term investments in the mining sector.

References

1. Santos, L., Soares, I., Mendes, C., Ferreira, P.: Real options versus traditional methods to assess renewable energy projects. Renew. Energy **68**, 588–594 (2014)
2. Espinoza, D., Morris, J.W.: Decoupled NPV: a simple, improved method to value infrastructure investments. Constr. Manag. Econom. **31**(5), 471–496 (2013)
3. Chang, C.Y.: A critical analysis of recent advances in the techniques for the evaluation of renewable energy projects. Int. J. Proj. Manag. **31**(7), 1057–1067 (2013)

4. Christensen, T.B., Bertelsen, T., Lorentzen, T.E., Lück, S., Maarbjerg, R.: Establishing the investment case—wind power, 27p. Deloitte (2015)
5. Wu, Z., Sun, H.: Behavior of Chinese enterprises in evaluating wind power projects: a review based on survey. Renew. Sustain. Energy Rev. **43**, 133–142 (2015)
6. Robichek, A.A., Myers, S.C.: Conceptual problems in the use of risk-adjusted discount rates? J. Financ. **21**(4), 727–730 (1966)
7. Espinoza, D.: Separating project risk from the time value of money: a step toward integration of risk management and valuation of infrastructure investments. Int. J. Proj. Manag. **32**(6), 1056–1072 (2014)
8. Espinoza, D., Rojo, J.: Using DNPV for valuing investments in the energy sector: a solar project case study. Renew. Energy **75**, 44–49 (2015)
9. Bock, K., Trück, S.: Assessing uncertainty and risk in public sector investment projects. Technol. Invest. **2**(2), 105–123 (2011)
10. Lüers, S., Wallasch, A.-K., Rehfeldt, K.: Kostensituation der Windenergie an Land in Deutschland—Update. Deutsche WindGuard, 65p. (2015)
11. Michelez, J., Rossi, N., Blazquez, R., Martin, J.M., Mera, E., Christensen, D., Peineke, C., Graf, K., Lyon, D., Stevens, G.: Risk quantification and risk management in renewable energy projects. Altran, IEA, 150p. (2011)

Part VIII
Game Theory and Experimental Economics

Impact of Non-truthful Bidding on Transport Coalition Profits

Jonathan Jacob and Tobias Buer

Abstract A coalition of freight carriers is considered which has to decide how to allocate a pool of transport requests among its members. The literature is aware of a number of solution approaches which usually assume truthful behavior of the freight carriers. However, the used negotiation protocols are mostly not proven to enforce truthful behavior. This paper gives some insights into the impact of non-truthful behavior via computational experiments. We solve the collaborative problem via a genetic algorithm (GA) which is operated by an auctioneer. The GA's individuals are allocations of requests to carriers. To calculate the fitness of an individual, the carriers bid on the allocations. Bidding below a carrier's true valuation could ceteris paribus increase its profits. However, understated valuations can influence the search process negatively, in particular when a favoured allocation is dismissed wrongly. It is shown via computational experiments that for six tested instances, bidding non-truthfully is individually, but not collectively, rational and results in a kind of prisoner's dilemma.

1 Introduction

A way multiple freight carriers can establish a coalition is through collaborative transportation planning. Members of horizontal coalitions (i.e. carriers) try to increase their profits by exchanging some of their transport requests [5]. Through the exchange, they expect to find better tour plans that increase service quality and provide a higher utilization of resources. Empirical results show that horizontal collaborations are seen as beneficial, however, opportunistic behavior is perceived as a threat [5]. One of the main questions members of transport coalitions face is how to allocate requests in a way that is profitable to the coalition.

J. Jacob (✉) · T. Buer
Computational Logistics Junior Research Group, University of Bremen,
Bibliothekstr. 1, 28359 Bremen, Germany
e-mail: jjacob@uni-bremen.de

T. Buer
e-mail: tobias.buer@uni-bremen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_28

Verdonck et al. [10] categorize request sharing techniques into either joint route planning or auction-based approaches. For joint route planning, a centralized decision maker is assumed who optimizes the decisions from the coalition's point of view. Auction-based approaches on the other hand consider that in most coalitions the carriers are autonomous, have therefore private information, and are self-interested. In an auction the requests are tendered, the carriers submit bids on the requests, and the auctioneer decides which bids win the auction. Individual carriers are responsible for their routing and valuation decisions [3, 4, 6, 11, 12] which appears to be a welcomed feature by many coalitions.

However, all of these recent studies assume truthful bidding. One reason may be that manipulations are non-trivial. The involved subproblems like the bid generation problem [4] or the winner determination problem [3] are hard to solve even without considering cheating. The Generalized Vickrey Auction, as an incentive compatible mechanism, is impracticable to apply for transport coalitions because of its high computational effort [3], its vulnerability to collusion by subsets of bidders, and its vulnerability to false-name bids [1].

In what follows, the transportation request assignment problem is introduced in Sect. 2 and a collaborative planning approach based on a genetic algorithm is presented in Sect. 3. In Sect. 4, the computational results on the impact of non-truthful bidding in a coalition of carriers are presented.

2 The Transportation Request Assignment Problem

Freight carriers collaborate by forming a coalition. A coalition is a set A of n self-interested and independent agents, here denoted as carriers. The coalition considers a set R of freight requests for servicing. The following *pairwise disjoint* subsets of R are relevant: Each carrier $a \in A$ holds an initial set of requests $I_a \subset R$. These are private and not for exchange. For any request in I_a, carrier $a \in A$ is obliged to personally fulfill it or pay a penalty when it is not fulfilled. Furthermore, a broker (or one or more shippers) offers the coalition a set $P \subset R$ of requests. The coalition can either accept all requests in the pool P or reject all of them. If P is accepted, the requests have to be serviced or penalty costs incur. Altogether, R is defined as $R := \bigcup_{a \in A} I_a \cup P$.

The coalition's goal is to maximize the profit by jointly servicing R, taking into account that I_a are private information ($a \in A$) and must not be revealed to other members of the coalition. The profit π_a of carrier $a \in A$ is defined in (1). It depends on a's allocated requests R_a (with $I_a \subseteq R_a \subseteq R$) and the winning bid price $b_a \in \mathbb{Z}$:

$$\pi_a(R_a, b_a) = p(R_a) + \frac{\sum_{i \in A} b_i}{n} - c(R_a) - b_a. \tag{1}$$

Profit π_a is *after* sharing the coalition's profit. The *income* of carrier a consists of $p(R_a)$, the sum of the paid prices for servicing requests in R_a (the price per request is given) and a's share in the coalition's profit. The coalition's profit is calculated from

the sum of the winning bid prices. It is assumed to be distributed among the carriers in equal shares, i.e. $\frac{\sum_{i \in A} b_i}{n}$. The *expenses* of carrier a consist of its winning bid price b_a (negative prices are possible) and the costs $c(R_a)$ of its tour plan for servicing R_a. Basically, these are made up of the fixed costs per tour, the tour length costs and in particular of the penalty costs when some requests in R_a are *failed* to be serviced. We assume all requests $r \in R$ are pickup-and-delivery requests with time windows [9]. In addition, for each request $r \in R$, a price p_r and a penalty cost for non-fulfillment q_r are given. Therefore, in order to calculate $c(R_a)$, a carrier has to solve the well-known and *NP*-hard pickup-and-delivery problem with time windows (PDPTW) to service the requests in R_a for minimum cost. The extension to the traditional PDPTW is that requests bear penalty costs if they are not fulfilled.

In order to agree on an allocation of the pooled requests to the carriers, the coalition has to solve the transportation request assignment problem (TRAP), given by formulas (2)–(6). The TRAP is basically a bi-level optimization problem based on the set partitioning problem. The task is to find a partition of the set of pooled requests P that consists of n subsets. Each subset P_a is assigned to exactly one carrier.

$$\max \sum_{a \in A} \pi_a(P_a \cup I_a, b_a) \tag{2}$$

$$\text{s.t.} \quad \bigcup_{a \in A} P_a = P \tag{3}$$

$$P_i \cap P_j = \emptyset \quad \forall i, j \in A, i \neq j \tag{4}$$

$$\sum_{a \in A} b_a \geq 0 \tag{5}$$

$$b_a \in \mathbb{Z} \quad \forall a \in A \tag{6}$$

The total profit (2) of the coalition should be maximized. All requests in the pool P have to be assigned to exactly one carrier, see (3) and (4). Furthermore, the sum of the carriers' bids has to be positive (5), otherwise it would be better for the coalition to reject P. In order to decide about the bid price b_a (6) on an allocation a carrier $a \in A$ has to calculate its marginal profits which requires solving the PDPTW.

3 A Genetic Algorithm with Bidding on Encoded Allocations

To solve the TRAP, Jacob and Buer [8] introduced a genetic algorithm (GA). Following [10], it is classified as an auction-based approach. It can be used by the mediator of the negotiation and enables collaboration of carriers while protecting private information to a large extent. The GA searches an allocation α, i.e., an assignment of all requests in P to carriers in A. To calculate the fitness of the individuals (i.e., the allocations) the carriers only revealed their ranking of the allocations; cost

information remained private which is an important feature. However, the surplus profit generated by the coalition was also unknown and could not be distributed between the members of the coalition.

In order to overcome this deficit, we now propose the carriers should evaluate an allocation via a—possibly negative—monetary value, i.e., they should bid on an allocation. One distinctive feature is that only bids on complete allocations are allowed; in contrast to bidding on subsets of the auctioned request which includes as special cases bids on single requests or bids on request bundles that are tours. Although a carrier has to reveal its price for an allocation, the revealed cost structure is much less detailed than, e.g., prices of sets of requests. In addition, the sum of the bid prices for an allocation is a nice indicator for the coalition's surplus profit. The main features of the GA are as follows.

Encoding of an individual. An individual of the GA represents an allocation α of requests to carriers. It is a sequence of carriers $a \in A$ of length $|P|$. Each position of the sequence represents a request in P. For example, the individual $\alpha = (3, 1, 3, 2)$ represents an allocation of four requests where carrier a_1 gets request 2, a_2 gets request 4, and carrier a_3 receives 1 and 3.

Fitness value. Different from [8], the fitness of an individual is calculated as the sum of the bid prices. A bid $b_a(\alpha)$ of carrier $a \in A$ may be positive or negative (see below).

Crossover and mutation. A standard 2-point-crossover is applied with a probability of 90%. Next, mutation is applied with a 30% probability. If an individual is mutated, the carrier at each position is replaced by a random one with a probability of 10%.

Truthful bidding on an allocation. In order to calculate the fitness of an individual, each carrier bids on an allocation. To start with and in line with the vast majority of the literature [3, 4, 6, 11, 12], we assume truthful bidding. Given an allocation α, each carrier $a \in A$ calculates its bid price $b_a(\alpha)$. To this end, each $a \in A$ solves a PDPTW taking into account its initial requests I_a and its additional requests P_a for each individual in each generation. Therefore, our mechanism is computationally challenging. We use an adaptive large neighborhood search [9] to generate a set of feasible tours; then we select a proper subset of tours via solving a set covering problem. From this solution we calculate the bid price b_a that equals the *marginal profit* resulting from servicing P_a in addition to I_a (taking penalty costs into account). Another benefit is that in this way the marginal profit of the coalition is revealed. Note, a bid on the same allocation in a later iteration may only be increased.

Incentives for non-truthful bidding on an allocation. Our GA-based auction protocol is not proven to enforce truthful bidding. On the winning allocation, a carrier $a \in A$ increases its profit by decreasing its bid price $b_a(\alpha)$. However, the lower the sum of the bids on an allocation are, the lower are its chances to get chosen.

The question is: how strong can a non-truthful carrier understate its preferences? Non-truthful bidding is implemented via calculating the bid price according to (7). The bid price is based on the concept of marginal profits. The income of all serviced requests is $p(P_a \cup I_a)$. The expenses of the serviced requests $c(P_a \cup I_a)$ are modified by the strategy δ_a, where $\delta_a = 0$ indicates truthful bidding and $\delta_a > 0$ indicates non-

truthful bidding. Without collaboration, the profit for servicing the initial requests I_a is denoted by $\overline{\pi}_a$. A negative bid price indicates the amount of money required to compensate the carrier for its losses due to collaborating.

$$b_a(\alpha) = p(P_a \cup I_a) - (1 + \delta_a) \cdot c(P_a \cup I_a) - \overline{\pi}_a \tag{7}$$

This non-truthful bidding scheme is implemented by a carrier consistently for all bids on all allocations throughout the complete negotiation process. The share of the coalition's profit is not considered. As the computational results in the next section show, this leads essentially to a prisoner's dilemma.

4 Results on Non-truthful Bidding and Discussion

For our tests, we created six Euclidean TRAP instances T2-1 to T2-6, each with two carriers ($n = 2$), twenty initial requests per carrier ($|I_a| = 20, a \in A$), and a pool of forty requests ($|P| = 40$). For each request $r \in R$ a price p_r was randomly chosen between 50 and 150, and a penalty cost q_r was randomly chosen between 200 and 300. Every time the GA presents an allocation α to a carrier, the carrier bids $b_a(\alpha)$ according to Eq. (7).

The parameter δ_a determines the bidding strategy of carrier $a \in A$. Truthful bidding is implied by $\delta_a = 0$. The greater δ_a, is the stronger a exaggerates its true costs and the lower are its bid prices. Table 1 shows the payoff matrix for $a = 1, 2$ and $\delta_a = 0.0, 0.35, 0.7$. The average *marginal* payoffs (Δ_1, Δ_2) for carrier 1 and carrier 2 over the instances T2-1 to T2-6 are given. Marginal payoff Δ_a is the profit of carrier $a \in A$ in the case of collaboration minus the profit without collaboration.

Assume now that each carrier knows those payoff matrices from observation and sees them as a means of predicting future payoffs. Then, the different values of δ_a can be interpreted as each carrier a's strategy in a game. Assuming rational behavior, carrier 1 will choose $\delta_1 = 0.35$ and carrier 2 will choose $\delta_2 = 0.35$ since this is the only Nash equilibrium [2]. But, if the carriers chose $\delta_1 = 0$ and $\delta_2 = 0$, they would both be better off. So apparently, collective rationality is not given. This holds also for the three carrier case, see working paper [7].

Table 1 Payoff matrix of averaged marginal profits (Δ_1, Δ_2)

		Carrier 2		
		$\delta_2 = 0.00$	$\delta_2 = 0.35$	$\delta_2 = 0.70$
Carrier 1	$\delta_1 = 0.00$	(1222, 1222)	(677, 1701)	(292, 1545)
	$\delta_1 = 0.35$	(1730, 765)	(917, 1016)	(435, 757)
	$\delta_1 = 0.70$	(1606, 329)	(396, 219)	(357, 292)

A possible instrument to induce truthful bidding is to introduce a deposit that each carrier has to pay in order to become a part of the coalition. If P gets successfully allocated, each carrier gets its deposit back. If, however, no feasible solution of the TRAP can be found, the deposits get returned unevenly: The higher a carrier's average bids are, the higher will be the amount it receives. How to choose the amount of the deposit and the exact mechanism to return the deposits in case no feasible solution is found may be the object of future research.

Acknowledgements The cooperative junior research group on Computational Logistics is funded by the University of Bremen in line with the Excellence Initiative of German federal and state governments.

References

1. Ausubel, L.M., Milgrom, P.: The lovely but lonely Vickrey auction. In: Cramton, P., Shoham, Y., Steinberg, R. (eds.) Combinatorial Auctions, pp. 17–40 (2006)
2. Avis, D., Rosenberg, G.D., Savani, R., Von Stengel, B.: Enumeration of Nash equilibria for two-player games. Econ. Theory **42**(1), 9–37 (2010). http://banach.lse.ac.uk
3. Berger, S., Bierwirth, C.: Solutions to the request reassignment problem in collaborative carrier networks. Transp. Res. Part E: Logist. Transp. Rev. **46**(5), 627–638 (2010)
4. Buer, T.: An exact and two heuristic strategies for truthful bidding in combinatorial transport auctions. Working paper, Computational Logistics, University of Bremen (2014). arXiv:1406.1928
5. Cruijssen, F., Cools, M., Dullaert, W.: Horizontal cooperation in logistics: opportunities and impediments. Transp. Res. Part E: Logist. Transp. Rev. **43**(2), 129–142 (2007)
6. Gansterer, M., Hartl, R.: Request evaluation strategies for carriers in auction-based collaborations. OR Spectr. **38**, 3–23 (2016)
7. Jacob, J., Buer, T.: Impact of non-truthful bidding on transport coalition profits. Bremen Computational Logistics Group Working Papers No. 3, University of Bremen. http://hdl.handle.net/10419/144761 (2016)
8. Jacob, J., Buer, T.: Population-based negotiation of contract clauses and transportation request assignments. IFAC-PapersOnLine **49**(12), 1862–1867 (2016). 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016, Troyes, France
9. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. **40**(4), 455–472 (2006)
10. Verdonck, L., Caris, A., Ramaekers, K., Janssens, G.K.: Collaborative logistics from the perspective of road transportation companies. Transp. Rev. **33**(6), 700–719 (2013)
11. Wang, X., Kopfer, H.: Collaborative transportation planning of less-than-truckload freight. OR Spectr. **36**(2), 357–380 (2014)
12. Ziebuhr, M., Kopfer, H.: Solving an integrated operational transportation planning problem with forwarding limitations. Transp. Res. Part E: Logist. Transp. Rev. **87**, 149–166 (2016)

Equilibrium Selection in Coordination Games: An Experimental Study of the Role of Higher Order Beliefs in Strategic Decisions

Thomas Neumann and Bodo Vogt

Abstract The equilibrium selection in games with multiple equilibria, such as coordination games, depends on one player's beliefs about the other player's behavior; as such, the outcome of the game depends on the players' expectations of one another's behavior. This study assessed the extent to which players' higher order beliefs influence the strategic choices they make during 2×2 coordination games. Using a quadratic scoring rule, the players' higher order beliefs about the choices their opponent would make were directly elicited in a laboratory experiment. The players' higher order beliefs were analyzed to ascertain the extent to which players' depth of thinking influenced their strategic decisions. In addition, this study focused on the question of whether the players update their beliefs to build higher order beliefs. The findings of the study revealed that the average participant operated on four steps of strategic depth. Higher order beliefs follow different patterns. In most cases, these contrast Bayesian updating.

1 Introduction

Consider the following symmetric 2×2 normal form coordination game, presented in Table 1. The game has two pure strategy Nash equilibria: a payoff dominant (A, A) and a risk dominant (B, B), which follows the two selection criteria introduced by Harsanyi and Selten [1]. This game also has one mixed strategy Nash equilibrium, where each player chooses A with a probability of 0.65.

The existing literature on equilibrium selection in coordination games is, broadly speaking, two folded. While many researchers, such as Harsanyi and Selten [1], argue for selecting the payoff dominant equilibrium, other authors attribute the greater weight to selecting the risk dominant equilibrium [2]. A high-level assessment of

T. Neumann (✉) · B. Vogt
Faculty of Economics and Management, Chair in Empirical Economics,
Otto-von-Guericke-University Magdeburg, P.O. Box 4120, 39016 Magdeburg, Germany
e-mail: t.neumann@ovgu.de

B. Vogt
e-mail: bodo.vogt@ovgu.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_29

Table 1 Game design

	A	B
A	(200, 200)	(0, 120)
B	(120, 0)	(150, 150)

these and other studies reveals that there is no common consensus on how equilibrium selection operates in coordination games [3].

While previous studies have examined how various factors, such as payoff structure, information structure, or risk attitude, influence players' decisions in coordination games [4–6], very few studies have focused on the impact that beliefs have on the strategies players adopt [7–9] and even fewer have examined the influence of higher order beliefs [10]. Recent studies have shown that players' stated first order beliefs can predict their decisions in a coordination game [3, 8, 11]. This study aimed to test the idea [11] that players' beliefs concerning other player's actions represent the key to understanding strategy selection in coordination games.

Since players simultaneously select a strategy in the game, they do not have access to objective information about the other player's behavior. Hence, they have to build expectations (or beliefs) regarding how they anticipate their opponent will act. To coordinate on an equilibrium, the players not only have to think about the strategy the other player will choose, but they also have to build expectations as to what the other player thinks, which action they will choose, etc.

Existing literature describes two main methods through which players beliefs can be elicited: direct or indirect elicitation. Trautmann and Kuilen [12] analyzed different belief elicitation methods and found that incentivized methods are better predictors of players' behavior. One of these incentivized methods is the quadratic scoring rule. This approach was employed in the current study to directly elicit players' higher order beliefs.

2 Depth of Reasoning, Higher Order Beliefs and Belief Updating

To study the depth of strategic thinking, we asked the players for the relevance of thinking about the strategy selection on a certain level, i.e., it is relevant for me, that it is relevant for you, that it is relevant for me, that..., to think about which strategy you chose. Using this design gave us an opportunity to reward the depth of thinking and accurateness of the matched players' relevance prediction without paying too much attention to the extent to which the strategy prediction was correct. While this would, undoubtedly, be of interest, it would open an additional strategic field of enquiry that would be difficult to isolate in the experimental analysis, especially given the structure of the payoff function.

Following considerations of former studies, we asked for a finite number of steps. Various studies have demonstrated that players use a limited number of steps of thinking [13]. Different studies corroborate that human beings tend to operate at only one or two levels of strategic depth [14]. Within the current study, we asked for eight steps, which should guarantee that all relevant steps were included. Our questions followed the scheme: *"It is relevant for me, that it is relevant for you, that it is relevant for me, that..., to think about which strategy you chose."*

In addition to the yes/no answer, the players were required to indicate by a number p ranging from 0–100, with 0 being not at all confident and 100 being fully confident, to denote the extent to which they were positive their prediction was correct. Standard game theory assumes indefinitely reasoning with a level of confidence at $p = 100$. To answer these questions, the players were rewarded according to a quadratic scoring rule that was adopted from a previous study [15]. This function is designed in such a manner that it is optimal for risk-neutral players to report their true beliefs [12]. The following functions were used in this experiment:

- if the matched players' relevance prediction is correct, the payoff is:

$$\frac{1}{7} \cdot \sum \left\{ 4 \cdot \left[1 - \left(1 - \frac{p}{100} \right)^2 \right] \right\} [in\ Euro], \text{and}$$

- if the matched players' relevance prediction is not correct, the payoff is:

$$\frac{1}{7} \cdot \sum \left\{ 4 \cdot \left[1 - \left(\frac{p}{100} \right)^2 - 0,3 \right] \right\} [in\ Euro].$$

With respect to players' loss aversion and the overweighting of losses [16], we designed the payoff function such that if the matched players' relevance predictions were not correct, the negative payoffs were much smaller than the positive payoffs that were rewarded for correct guesses. Furthermore, we used an average function to ensure all steps of thinking were relevant for the payoffs. Given this, reporting $p = 35$ on each step guaranteed a riskless payoff of 2.31 Euro.

In terms of depth of thinking, we were interested in how players modified their beliefs as the game progressed. Bayesian updating is potentially the most common theoretical concept related to this idea. The Bayesian rule of belief updating follows the intuition that the deeper one thinks, the lower the degree of confidence that one attaches to a certain event. In other words, as a decision maker progresses through consecutive steps of thinking, the degree of uncertainty will increase. The levels of certainty in the first two steps determine the maximum level of certainty in the following step. In our experiment, we considered any probability that was equal or smaller to the theoretical Bayesian probability to be the result of Bayesian updating.

3 Research Hypotheses and Experimental Procedure

Various studies have demonstrated that human beings typically tend to operate on only one or two levels of strategic depth in terms of the depth of reasoning [13, 14]. The current research sought to assess whether the players' depth of thinking influenced the strategic decisions they made in the game. To conclude and operationalize this question, we formulated our first hypothesis:

H1: The players' depth of thinking influences their strategy selections in the game.

Our experiment focused on depth of thinking. We were interested in the numbers of steps the players thought through and how their beliefs updated as the game progressed. The concept of Bayesian updating was used as a reference model. Many studies have pointed out that players do not behave as so-called "perfect Bayesians". Moreover, they are often not even close [17]. With respect to this finding, we formulated our second hypothesis:

H2: The players update their beliefs according to the Bayes rule to form higher order beliefs.

To test our hypotheses, we ran an experiment in the MaXLab, the experimental laboratory of the University of Magdeburg. The participants consisted of students from various faculties of the university. We ran our experiment over six sessions with groups of six subjects each. A program that was implemented in z-Tree was employed for the computerized parts of the research [18].

The participants were not permitted to communicate with each other at any point during the experiment. They also did not receive any information about their payoffs or the behavior of their partners. In total, it was possible for the participants to earn a maximum of 12 Euro. The experiment provided a riskless payoff of 7.02 Euro.

Before commencing the experiment, the participants were required to answer a series of questions that were designed to verify that they understood the meaning of the different steps of thinking. We also presented a computer screen to the participants on which they could try different probabilities to develop a better understanding of the payoff function employed within the study. After all the participants had given the correct answers to the questions, it was assumed that they fully understood the game, and the experiment commenced.

We ran our experiment over two rounds. In each round, two players were randomly matched to play the coordination game explained in Sect. 1. We used a matching mechanism that guaranteed that the participants did not interact with each other in previous rounds and that the matched participants were not matched with the same other participant in a previous round.

The decision for one of the two possible actions was equal to step zero in terms of the depth of thinking requested. As pointed out in Sect. 2, we asked for the relevance of thinking on eight steps. For that purpose, we used a questionnaire with a table on it. On each of the eight steps of strategic depth, the participants were required to disclose whether they perceived this step to be relevant by marking it with a cross to denote yes or no. The players were also required to indicate their level of confidence in their prediction as explained in Sect. 2.

4 Results

Table 2 presents the strategy selections observed in the coordination game across the two rounds of the experiment. The distribution of these strategy selections across both rounds shows no significant difference.

Within the experiment, the average participant operated on four steps of strategic depth in both rounds. The subjects were divided into two groups according to the strategy selections of the participants. Table 3 presents the medians of the depth of thinking of the participants in each of the two groups. In the first round, the 24 players who selected the risky strategy (A) operated on five steps of strategic depth, whereas the other 12 players who selected strategy (B) operated on only three steps. Interestingly, this approach was not replicated in the second round and players in both groups operated on four steps.

Thus, we concluded that, in the second round, the depth of thinking did not influence the players' strategy selection (Wilcoxon-Test, 5%-level) and H1 was rejected for round two. In the first round, the medians of the depth differed.

To assess Bayesian belief updating, we normalized the elicited beliefs by using the complementary probability. In our experiment, only four players in each round behaved as so-called Bayesians. One of these four followed the rational solution and reported the maximum probability $p = 100$ as level of confidence for all eight steps. The majority of the participants formed beliefs according to various patterns (one-sided Binomial-Test, 5%-Level), which were, in most cases, in contrast to the concept of Bayesian belief updating. As such, H2 was rejected.

Table 2 Distribution of the strategy selections

Strategy selection—no. of players		
	Round 1	Round 2
Strategy A (payoff dominant)	24	22
Strategy B (risk dominant)	12	14

Table 3 Strategy selection and median of the depth of thinking

Strategy selection—depth of reasoning (median)		
	Round 1	Round 2
Strategy A (payoff dominant)	5	4
Strategy B (risk dominant)	3	4

5 Conclusion

The starting point of this study was the idea that players' beliefs determine the strategies they employ in symmetric 2×2 coordination games. The research specifically focused on the question of whether the participants' updated their beliefs to build higher order beliefs. In addition, we analyzed if there was any influence of the depth of thinking on the strategic decision in the coordination game. On average, subjects reported four steps as relevant, which is equal to the depth of thinking. While we found different depth of thinking in accordance with the strategy selected in the first round, we did not find these differences in the second round. To study the belief updating mechanism performed by the participants, we used the Bayesian updating as the reference model. Our results revealed that only 4 of 36 subjects behaved like a Bayesian player.

References

1. Harsanyi, J.C., Selten, R.: A General Theory of Equilibrium Selection in Games. The MIT Press (1988)
2. Carlsson, H., Van Damme, E.: Global games and equilibrium selection. Econom. J. Econom. Soc. **61**(5), 989–1018 (1993)
3. Berninghaus, S.K., Haller, S., Krüger, T., Neumann, T., Schosser, S., Vogt, B.: Risk attitude, beliefs, and information in a corruption game—an experimental analysis. J. Econ. Psychol. **34**, 46–60 (2013)
4. Schmidt, D., Shupp, R., Walker, J.M., Ostrom, E.: Playing safe in coordination games: the roles of risk dominance, payoff dominance, and history of play. Games Econ. Behav. **42**(2), 281–299 (2003)
5. Cabrales, A., Nagel, R., Armenter, R.: Equilibrium selection through incomplete information in coordination games: an experimental study. Exp. Econ. **10**(3), 221–234 (2007)
6. Heinemann, F., Nagel, R., Ockenfels, P.: Measuring strategic uncertainty in coordination games. Rev. Econ. Stud. **76**(1), 181–221 (2009)
7. Rey-Biel, P.: Equilibrium play and best response to (stated) beliefs in normal form games. Games Econ. Behav. **65**(2), 572–585 (2009)
8. Neumann, T., Vogt, B.: Do players beliefs or risk attitudes determine the equilibrium selections in 2×2 coordination games? FEMM Working Paper Series, No. 09024 (2009)
9. Büyükboyacı, M.: Risk attitudes and the stag-hunt game. Econ. Lett. **124**(3), 323–325 (2014)
10. Kneeland, T.: Coordination under limited depth of reasoning. University of British Columbia Working Paper (2012)
11. Wang, S.: The role of risk aversion and cautiousness in belief formation. University of Connecticut (2015)
12. Trautmann, S.T., Kuilen, G.: Belief elicitation: a horse race among truth serums. Econ. J. **125**(589), 2116–2135 (2015)
13. Colman, A.M.: Depth of strategic reasoning in games. Trends Cogn. Sci. **7**(1), 2–4 (2003)
14. Camerer, C.F., Ho, T.-H., Chong, J.-K.: A cognitive hierarchy theory of one-shot games. Q. J. Econ. **119**(3), 861–898 (2004)
15. Nyarko, Y., Schotter, A.: An experimental study of belief learning using elicited beliefs. Econom. J. Econom. Soc. **70**(3), 971–1005 (2002)
16. Kahneman, D., Tversky, A.: Prospect theory: An analysis of decision under risk. Econometrica: Journal of the Econometric Society **47**(2), 263–291 (1979)

17. Charness, G., Levin, D.: When optimal choices feel wrong: a laboratory study of bayesian updating, complexity, and affect. Am. Econ. Rev. **95**(4), 1300–1309 (2005)
18. Fischbacher, U.: z-Tree: Zurich toolbox for ready-made economic experiments. Exp. Econ. **10**(2), 171–178 (2007)

Designing Inspector Rosters with Optimal Strategies

Stephan Schwartz, Thomas Schlechte and Elmar Swarat

Abstract We consider the problem of enforcing a toll on a transportation network with limited inspection resources. We formulate a game theoretic model to optimize the allocation of the inspectors, taking the reaction of the network users into account. The model includes several important aspects for practical operation of the control strategy, such as duty types for the inspectors. In contrast to a formulation in Borndörfer et al. (Networks, 65, 312–328, [1]) using flows to describe the users' strategies we choose a path formulation and identify dominated user strategies to significantly reduce the problem size. Computational results suggest that our approach is better suited for practical instances.

1 Introduction

In the past years, a lot of work has been done in the application of game theoretic models to real-world security problems. These applications range from airport security [4] over protection of wildlife reserves [4] to toll (or fare) control in transportation networks [1, 3]. In [4], the authors give an overview on projects with security games where mostly no network structures are considered. Models for fare evasion in public transport are studied in [3], but the work is focussed more on theoretical results than on practical issues. For practical operation it is important to include the notion of duties for inspection units and the concept of control areas as subparts of the network where controls can be conducted. Both of these extensions are taken into account in [1] where a game theoretic formulation for the enforcement of a toll on a transportation network is studied.

S. Schwartz · T. Schlechte · E. Swarat (✉)
Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
e-mail: swarat@zib.de

S. Schwartz
e-mail: schwartz@zib.de

T. Schlechte
e-mail: schlechte@zib.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_30

In this paper we reformulate the toll enforcement problem of [1]. By identifying dominated user strategies, we significantly reduce the size of the presented MIP and LP formulations to compute the control strategy in a Stackelberg and Nash equilibrium, respectively. Computational results for real-world instances show that the new approach outperforms the existing formulation.

2 The Toll Enforcement Game

We consider a user network $G_0 = (V_0, E_0)$ with nodes V_0 and directed arcs E_0 with costs $c_e \geq 0$. For a given time interval, typically one day or one week, we consider an equidistant time discretization $\mathscr{T} = \{0, \dots, T-1\}$. A time-expanded graph $G = (V, E)$ is constructed by adding a copy of G_0 for every time window $t \in \mathscr{T}$. In addition, we are given k commodities $(s_i, t_i, d_i) \in V \times V \times \mathbb{N}$ describing the number of users d_i travelling from s_i to t_i. We make the simplifying assumption that every network user starts and ends his trip within the same time window. Every user going from s_i to t_i is supposed to pay a toll (or fare) of τ_i. In contrast to the driving costs c_e on arc e, the users can decide not to pay the toll and risk a fine $f \gg \tau_i$ if caught evading. In order to enforce the toll, a number of κ inspection units can be allocated throughout the network. However, the possible distributions of the inspectors are subject to a number of spatial, temporal and legal constraints which will be specified later. In the following, we describe a game between the network users and the inspectors concerning the users' payment of the toll. While there is one player for every origin-destination pair (s_i, t_i), the inspectors are aggregated as one player choosing a joint control strategy.

Users' strategies: The set Σ_i of pure strategies of player i can be divided into toll paying strategies Σ_i^{pay} and toll evading strategies Σ_i^{ev}. If we consider the user network as the toll evading network where no toll is paid, we have

$$\underset{i}{\overset{ev}{\Sigma}} = \{P \mid P \text{ is an } s_i\text{-}t_i\text{-path in } G\}.$$

If player i decides to pay the toll τ_i she will take a shortest s_i-t_i-path with respect to the travel costs c. Considering the payoff functions we can assume that there is a single toll paying strategy for player i and we write $\Sigma_i^{pay} = \{\sigma_i^{pay}\}$.

With the mixed strategy $x^i = (x_0^i, x_1^i, \dots, x_{k_i}^i)$ we say that player i commits to σ_i^{pay} with probability x_0^i and to $P_j^i \in \Sigma_i^{ev}$ with probability x_j^i. The joint strategy of the users is denoted by $x = (x^1, \dots, x^k)$.

We would like to point out that [1] uses an equivalent formulation which describes the toll paying strategy of player i as an s_i-t_i-path in an adopted user network. Consequently, every mixed strategy of player i can be seen as an s_i-t_i-flow of unit value in this network.

Inspector's strategies: The spatio-temporal allocation of the inspectors is done by assigning duties to control areas. A duty can start at the beginning of every time window $t \in \mathscr{T}$ and is scheduled for a fixed number L of consecutive time windows. The control takes place on given *control areas* $\mathscr{A} = \{a_1, \dots, a_m\}$ with a predefined adjacency of control areas $A' \subseteq \mathscr{A}^2$. For every part $l = 1, \dots, L$ of the duty the inspector can switch from a_i to a_j iff $(a_i, a_j) \in A'$.

We define the set $\mathscr{D}$ of *control duties* to be

$$\mathscr{D} := \left\{ \left(t, (a^1, a^2, \dots, a^L)\right) \in \mathscr{T} \times \mathscr{A}^L \;\middle|\; (a^i, a^{i+1}) \in A' \right\}.$$

The set of the inspector's pure strategies can then be described as $\{C \subseteq \mathscr{D} \mid |C| \leq \kappa\}$.

In the following, we construct a duty graph $D = (W,\ A)$ to obtain a more elegant representation of the inspector's strategy set Σ_{insp}. For every time window t, every duty part l and every control area s_i we have a *control node* $(t, l, a_i) \in W$. If $l < L$ we introduce an arc $\left((t, l, a_i), (t, l+1, a_j)\right)$ iff $(a_i, a_j) \in A'$. Now we add additional nodes t^s and t^t for every time window t and insert arcs $\left(t^s, (t, 1, a_i)\right)$ and $\left((t, L, a_i), t^t\right)$ for every $a_i \in \mathscr{A}$. Finally, we introduce a super source d^s and a super sink d^t and arcs (d^s, t^s) and (d^t, t_d) for all $t \in \mathscr{T}$.

We can observe that there is a one-to-one correspondence between control duties and d^s-d^t-paths in D. The set of strategies for the inspection player can thus be formulated as

$$\Sigma_{insp} := \{p \mid p \text{ is a } d^s\text{-}d^t\text{-flow of value } \leq \kappa \text{ in } D\}.$$

For a given strategy $p \in \Sigma_{insp}$, the control intensities $q = (q_e)$ on arcs E of the user network G can be obtained by a given linear transformation, i.e. $q = Tp$. The induced control intensity q_e can be interpreted as the expected number of controls on arc $e \in E$. We follow the notation of [1] and define the set $\mathscr{Q}$ of induced control intensities q on G to be

$$\mathscr{Q} := \{Tp \mid p \in \Sigma_{insp}\}.$$

Payoffs: While the inspection player wants to maximize his total income, the users aim to minimize their total costs consisting of travel costs and toll costs or expected fine. The travel costs of player i choosing strategy $\sigma \in \Sigma_i$ are denoted by c_i^σ. If $\sigma = \sigma_i^{pay}$ then c_i^σ is the length of a shortest s_i-t_i-path with respect to c. For $\sigma = P \in \Sigma_i^{ev}$ we have $c_i^\sigma = \sum_{e \in P} c_e$.

If player i chooses strategy σ_i^{pay}, the player's and inspector's payoffs are independent of the chosen control strategy $p \in \Sigma_{insp}$. Then, the total costs of player i are

$$-\pi_i(p, \sigma_i^{pay}) := c_i^{\sigma_i^{pay}} + \tau_i,$$

while the inspector's profit from player i in this case is $\pi_{insp}^i(p, \sigma_i^{pay}) := \tau_i$.

Let us now assume, that player i chooses the evading strategy $P \in \Sigma_i^{ev}$ while the inspector plays $p \in \Sigma_{insp}$. With the induced control intensities $q = Tp$ on G we have $-\pi_i(p, P) := c_i^P + \sum_{e \in P} fq_e$ where the first term accounts for travel costs while the second term is the expected fine. Accordingly, the inspector's gain from player i is $\pi_{insp}^i(p, P) := \sum_{e \in P} fq_e$. Note that we use a simplified formula for the expected fine where we assume that evaders can be fined several times. However, our results show that the probability of being controlled more than once is very small for a reasonable number of controllers. With the above formula we also assume that the payoff for player i does not depend on the actions of the other users as we take no congestion effects into account. Given the control strategy p and the joint users' strategy $x = (x^1, \dots, x^k)$, we have

$$\pi_i(p, x^i) = x_0^i \, \pi_i(p, \sigma_i^{pay}) + \sum_{j=1}^{k_i} x_j^i \, \pi_i(p, P_j^i)$$

$$\text{and} \quad \pi_{insp}(p, x) = \sum_{i=1}^{k} \left(x_0^i \, \pi_{insp}^i(p, \sigma_i^{pay}) + \sum_{j=1}^{k_i} x_j^i \, \pi_{insp}^i(p, P_j^i) \right).$$

We denote by $BR_i(p)$ the set of best responses of player i to the control strategy p, i.e. $BR_i(p) := \arg\max_{x^i} \pi_i(p, x^i)$.

3 Computing Equilibria

Stackelberg Equilibrium: In most security games and fare evasion models the classical concept of Stackelberg equilibria is applied. A Stackelberg game is a bilevel game where the players are divided into leaders and followers. First, each leader (in our case the inspection player) commits to a strategy, then the followers choose a strategy after observing the leaders' strategy. Let p be a control strategy and x be a joint strategy of the users, then

$$(p, x) \text{ is a strong Stackelberg equilibrium } :\iff (p, x) \in \underset{(\tilde{p}, \tilde{x}) \,:\, \tilde{x}^i \in BR_i(\tilde{p})}{\arg\max} \pi_{insp}(\tilde{p}, \tilde{x}).$$

Note that the notion of strong Stackelberg equilibria implies that the followers break ties in favor of the leader. As a consequence, we only need to consider pure strategies of the followers [2]. While the existence of a strong Stackelberg equilibrium is always guaranteed, the respective optimization problem is NP-hard in general [2].

In the following we present a mixed integer program (MIP) to compute a leader strategy of a Stackelberg equilibrium for the toll enforcement game.

$$\max_{q,y,\mu} \quad \sum_i d_i \left(y_i - \sum_{\sigma \in \Sigma_i} \mu_i^\sigma \, c_i^\sigma \right) \tag{1a}$$

$$\text{s.t.} \quad 0 \le c_i^{\sigma_i^{pay}} + \tau_i - y_i \le M \left(1 - \mu_i^{\sigma_i^{pay}}\right) \qquad \forall i \tag{1b}$$

$$0 \le c_i^P + \sum_{e \in P} f q_e - y_i \le M \left(1 - \mu_i^P\right) \qquad \forall P \in \Sigma_i^{ev} \; \forall i \tag{1c}$$

$$\sum_{\sigma \in \Sigma_i} \mu_i^\sigma = 1 \qquad \forall i \tag{1d}$$

$$\mu_i^\sigma \in \{0,1\} \qquad \forall \sigma \in \Sigma_i \; \forall i \tag{1e}$$

$$q \in \mathcal{Q} \tag{1f}$$

The objective (1a) is to maximize the inspector's income. This can be done by considering the total costs y_i of an optimal strategy of player i subtracted by her travel costs. The costs y_i are bounded from above by the costs of the toll paying strategy (1b) and the costs of any evasion strategy (1c). The binary variable μ_i^σ indicates if $\sigma \in \Sigma_i$ is a best response to the control q. Constraints (1b) and (1c) also guarantee that $\mu_i^\sigma = 0$ if σ is not a best response for player i. Equation (1d) and the second term in the objective function make sure that each follower breaks ties in favor of the leader. Finally, in (1f) we force q to be induced by a control flow $p \in \Sigma_{insp}$.

Nash Equilibrium: We also study the Nash equilibria of the toll enforcement game which can be derived for the present case as follows:

$$(p,x) \text{ is a Nash equilibrium } :\iff p \in \operatorname*{arg\,max}_{\tilde{p} \in \Sigma_{insp}} \pi_{insp}(\tilde{p},x) \text{ and } x^i \in BR_i(p).$$

The existence of a Nash equilibrium in the toll enforcement game is guaranteed and an optimal strategy for the inspection player can be computed by linear programming due to the following important result from [1]: Let x be a joint mixed strategy for the users, then

$$p \in \operatorname*{arg\,max}_{\tilde{p} \in \Sigma_{insp}} \pi_{insp}(\tilde{p},x^i) \iff p \in \operatorname*{arg\,max}_{\tilde{p} \in \Sigma_{insp}} \sum_{i=1}^{k} -\pi_i(\tilde{p},x^i).$$

Therefore, the inspection player aims to maximize the costs of the users in a Nash equilibrium and his optimal strategy can be computed with the following linear program (LP):

$$\max_{q,r} \quad \sum_i d_i \, r_i \tag{2a}$$

$$\text{s.t.} \quad r_i \le c_i^{\sigma_i^{pay}} + \tau_i \qquad \forall i \tag{2b}$$

$$r_i \le \sum_{e \in P} c_e + f q_e \qquad \forall P \in \Sigma_i^{ev} \; \forall i \tag{2c}$$

$$q \in \mathcal{Q} \tag{2d}$$

Due to (2a) the inspection player aims to maximize the total costs of the users. The costs for player i described by r_i are bounded from above by the costs of the toll paying strategy (2b) and also by the costs of her evading strategies (2c). Again, we force q to be induced by a control flow $p \in \Sigma_{insp}$ (2d).

Dominated strategies: The number Σ_i^{ev} of toll evading strategies for player i is potentially huge compared to the size of the network. It is well known that the number of paths in a graph can be exponential in the number of edges. To avoid a potentially great number of constraints (1c) and (2c) the authors of [1] use a flow formulation to describe the users' strategies.

In practice however, user networks are normally sparse and there are not a huge number of possible user paths, especially if we exclude dominated strategies. In those networks, the travel costs represent the largest share of the user's total costs while toll costs or expected fines are secondary. As a result, the travel costs of most s_i-t_i-paths exceed the toll paying costs of $c_i^{\sigma_i^{pay}} + \tau_i$. A great number of strategies $P_j^i \in \Sigma_i^{ev}$ are thus dominated by the honest strategy σ_i^{pay}.

We use a preprocessing algorithm to compute the honest costs for every player i and apply a modified version of Yen's k-shortest path algorithm [5] to find the s_i-t_i-paths in G with length $\leq c_i^{\sigma_i^{pay}} + \tau_i$ and thereby build the set Σ_i^{ev}.

4 Computational Results

We applied the presented approaches to three real-world instances of the German motorway network. The instances were provided by the federal office for goods transport who is responsible for the truck toll enforcement on German motorways. The commodities are based on historical data and we schedule the duties for an exemplary week with 4 h time windows and duties with two parts. The optimiza-

Table 1 Computation of the inspector's strategy in a Nash equilibrium for three real-world instances with $|\mathscr{T}| = 168$ and $L = 2$. We compare the flow formulation of (2) taken from [1] to the presented path formulation with non-dominated strategies. Computation time includes preprocessing, building and solving time, RAM shows the maximum memory usage during the computation

Instance	$\lvert V_0\rvert$	$\lvert E_0\rvert$	k	# rows in reduced LP	# columns in reduced LP	Computation time in s	RAM
I1_flow	112	220	118,917	929,445	510,453	482	4.6 GB
I1_paths				67,299	75,194	29	0.4 GB
I2_flow	196	394	220,204	2,997,920	1,569,627	17,095	23.0 GB
I2_paths				157,870	167,317	214	3.7 GB
3_flow	319	672	365,603	7,593,778	3,718,269	–	killed
I3_paths				270,799	235,759	338	7.7 GB

tion was run on a Linux PC (3.6 GHz, 8 cores, 32 GB RAM) and we used CPLEX as an LP and MIP solver.

We also computed Stackelberg equilibria for the above instances using the path formulation in the MIP (1). The computation time and RAM usage were similar to the respective Nash equilibria. Noting that the computation of a Stackelberg equilibrium is at least as hard as computing a Nash equilibrium, we expect the results from Table 1 to carry over.

References

1. Borndörfer, R., Buwaya, J., Sagnol, G., Swarat, E.: Network spot-checking games: theory and application to toll enforcing in transportation networks. Networks **65**, 312–328 (2015)
2. Conitzer, V., Sandholm, T.: Computing the optimal strategy to commit to. In: Proceedings of the ACM Conference on Electronic Commerce (ACM-EC), pp. 82–90 (2006)
3. Correa, J., Harks, T., Kreuzen, V., Matuschke, J.: Fare Evasion in Transit Networks (2014). arXiv:1405.2826
4. Nguyen, T., Kar, D., Brown, M., Sinha, A., Jiang, A., Tambe, M.: Towards a science of security games. In: Toni, B. (ed.) New Frontiers of Multidisciplinary Research in STEAM-H (2016)
5. Yen, J.: Finding the k shortest loopless paths in a network. Manage. Sci. **17**, 712–716 (1971)

Part IX
Graphs and Networks

A Mixed-Integer Nonlinear Program for the Design of Gearboxes

Lena C. Altherr, Bastian Dörig, Thorsten Ederer, Peter F. Pelz, Marc E. Pfetsch and Jan Wolf

Abstract Gearboxes are mechanical transmission systems that provide speed and torque conversions from a rotating power source. Being a central element of the drive train, they are relevant for the efficiency and durability of motor vehicles. In this work, we present a new approach for gearbox design: Modeling the design problem as a mixed-integer nonlinear program (MINLP) allows us to create gearbox designs from scratch for arbitrary requirements and—given enough time—to compute provably globally optimal designs for a given objective. We show how different degrees of freedom influence the runtime and present an exemplary solution.

1 Introduction

A gearbox transfers power from the input shaft (driven by the motor) to the output shaft (connected to the differential) by engaging gear wheels, cf. [1]. By changing the size and interconnection of these components, the total transmission ratios, and thus

L.C. Altherr · T. Ederer (✉) · P.F. Pelz
Department of Mechanical Engineering, Technische Universität Darmstadt,
Otto-Berndt-Str. 2, 64287 Darmstadt, Germany
e-mail: thorsten.ederer@fst.tu-darmstadt.de

L.C. Altherr
e-mail: lena.altherr@fst.tu-darmstadt.de

P.F. Pelz
e-mail: peter.pelz@fst.tu-darmstadt.de

B. Dörig · T. Ederer · M.E. Pfetsch
Department of Mathematics, Technische Universität Darmstadt,
Dolivostr. 15, 64293 Darmstadt, Germany
e-mail: doerig@mathematik.tu-darmstadt.de

M.E. Pfetsch
e-mail: pfetsch@mathematik.tu-darmstadt.de

J. Wolf
Department of Economics, Universität Siegen, Unteres Schloß 3, 57072 Siegen, Germany
e-mail: jan.wolf@uni-siegen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_31

the resulting torque and angular velocity of the output shaft can be set. However, each design comes with its own issues: higher weight, worse efficiency, or an impractical shape. Therefore, automobile manufacturers are confronted with a complex design problem.

2 Technical Application

We deal with the optimal design of so-called dual-clutch transmissions. These modern transmissions allow to change gears without interruption of traction. This is realized by using two input shafts which can separately be rotated and clutched to the motor. While even numbered gears are assigned to one input shaft, odd numbered gears are assigned to the other input shaft. To save space, the two input shafts are designed as a long full shaft fitted inside a shorter hollow shaft.

In terms of wear and noise it is advantageous to always engage the gear wheels. To be able to do so without blockage, the wheels are pivot-mounted on the shafts, i.e., they can rotate independently from another, and sliding sleeves are used to fix a wheel to its respective shaft. One sliding sleeve placed between two wheels can synchronize either of them, but not both simultaneously. Due to the complexity of such synchronization systems, minimizing the number of sliding sleeves has high priority. To save a synchronizer, selected gear wheels can also be fixed on the shaft.

3 Gearbox Design via MINLP

While individual aspects of the configuration of transmissions have been studied with the aid of mathematical optimization methods (for an overview see e.g. [2]), a holistic approach could lead to even better solutions. In a first step towards this goal, we present a mixed-integer nonlinear program (MINLP) for finding an optimal gearbox design.

In our model capital letters denote sets or parameters, small letters denote continuous decision variables or indices, and greek letters denote binary decision variables. Table 1 gives an overview of all decision variables. Three index sets are used: The set of gears $G = \{-1, 1, 2, \dots\}$, including one reverse gear -1, the set of planes in axial direction $P = \{1, 2, \dots\}$, and the set of the two drive shafts $D = \{1, 2\}$. A component's position is characterized by a plane and a shaft index.

$$\text{minimize} \qquad h + W_1 \cdot \sum_{p \in P} \sum_{d \in D} \sigma_{p,d} + W_2 \cdot \sum_{p \in P} \zeta_p \tag{1}$$

subject to

$$K_g^{\min} \le i_g \cdot \sum_{d \in D} j_d \cdot \sum_{p \in P} \xi_{g,p,d} \le K_g^{\max} \qquad \forall g \in G \tag{2}$$

Table 1 Decision variables of the mixed-integer nonlinear program

Var.	Description	Domain
i_g	Pre-transmission of gear g	$[I^{\min}, I^{\max}]$
j_d	Post-transmission from drive shaft d to the output shaft	$[J^{\min}, J^{\max}]$
r_g	Gear wheel radius on the input shaft belonging to gear g	$[R^{\min}, R^{\max}]$
s_g	Gear wheel radius on the drive shaft belonging to gear g	$[S^{\min}, S^{\max}]$
t_p	Gear wheel radius on the input shaft in plane p	$[R^{\min}, R^{\max}]$
$u_{p,d}$	Gear wheel radius on drive shaft d in plane p	$[S^{\min}, S^{\max}]$
v_d	Maximum gear wheel radius on drive shaft d	$[S^{\min}, S^{\max}]$
y_d	Gear wheel radius on drive shaft d of the final drive	$[Y^{\min}, Y^{\max}]$
z	Gear wheel radius on the output shaft of the final drive	$[Z^{\min}, Z^{\max}]$
$\xi_{g,p,d}$	Indicator whether gear g is realized on drive shaft d in plane p	$\{0, 1\}$
$\gamma_{p,d}$	Indicator whether any gear is realized on drive shaft d in plane p	$\{0, 1\}$
$\delta_{g,p}$	Indicator whether gear g is realized on any drive shaft in plane p	$\{0, 1\}$
ζ_p	Indicator whether any gear is realized on any drive shaft in plane p	$\{0, 1\}$
$\sigma_{p,d}$	Indicator whether a sleeve is located on drive shaft d in plane p	$\{0, 1\}$
ϕ_p	Indicator whether the input shaft is a full shaft in plane p	$\{0, 1\}$
a_d	Distance between input shaft and drive shaft d	$[A^{\min}, A^{\max}]$
b_d	Distance between drive shaft d and output shaft	$[B^{\min}, B^{\max}]$
c	Distance between the drive shafts	$[C^{\min}, C^{\max}]$
h	Pseudo-height of the gearbox	$[H^{\min}, H^{\max}]$

$$r_g \cdot i_g = s_g \cdot \operatorname{sign}(g) \qquad \forall g \in G \tag{3}$$

$$y_d \cdot j_d = z \qquad \forall d \in D \tag{4}$$

$$\delta_{g,p} = \sum_{d \in D} \xi_{g,p,d} \qquad \forall g \in G,\, p \in P \tag{5}$$

$$1 = \sum_{p \in P} \delta_{g,p} \qquad \forall g \in G \tag{6}$$

$$\gamma_{p,d} = \sum_{g \in G} \xi_{g,p,d}\,, \quad \zeta_p \geq \gamma_{p,d} \qquad \forall p \in P,\, d \in D \tag{7}$$

$$\zeta_p \leq \sum_{d \in D} \gamma_{p,d} \qquad \forall p \in P \tag{8}$$

$$\gamma_{p,d} + \sigma_{p,d} \leq 1\,, \quad \gamma_{p,d} \leq \sum_{\substack{p' \in P \\ |p'-p|=1}} \sigma_{p',d}\,, \quad \xi_{-1,p,d} \leq \gamma_{p,3-d} \qquad \forall p \in P,\, d \in D \tag{9}$$

$$\phi_p \geq \phi_{p+1} \qquad \forall p \in P,\, p < |P| \tag{10}$$

$$\delta_{g,p} \leq \begin{cases} \phi_p & \text{if } g \text{ is odd} \\ 1 - \phi_p & \text{if } g \text{ is even} \end{cases} \qquad \forall p \in P,\, g \in G, g > 0 \tag{11}$$

$$\delta_{-1,p} \leq 1 - \phi_p \qquad \forall p \in P \tag{12}$$

$$r_g = \sum_{p \in P} t_p \cdot \delta_{g,p}\,, \quad s_g = \sum_{p \in P} \sum_{d \in D} u_{p,d} \cdot \xi_{g,p,d} \qquad \forall g \in G \tag{13}$$

$$t_p \geq R^{\text{full}} \cdot \phi_p + R^{\text{hollow}} \cdot (1 - \phi_p) + R^{\max} \cdot \zeta_p \qquad \forall p \in P \tag{14}$$

$$t_p \leq R^{\text{full}} \cdot \phi_p + R^{\text{hollow}} \cdot (1 - \phi_p) + R^{\max} \cdot \zeta_p \qquad \forall p \in P \tag{15}$$

$$u_{p,d} \geq S^{\min} \cdot (1 - \sigma_{p,d} - \gamma_{p,d}) + S^{\text{sync}} \cdot \sigma_{p,d} + S^{\min} \cdot \gamma_{p,d} \qquad \forall p \in P,\, d \in D \tag{16}$$

$$u_{p,d} \leq S^{\min} \cdot (1 - \sigma_{p,d} - \gamma_{p,d}) + S^{\text{sync}} \cdot \sigma_{p,d} + S^{\max} \cdot \gamma_{p,d} \qquad \forall p \in P,\, d \in D \tag{17}$$

$$v_d \geq u_{p,d} \qquad \forall p \in P, d \in D \tag{18}$$

$$a_d \geq t_p + u_{p,d} + \frac{1}{2} \cdot Q \cdot (1 - \gamma_{p,d}) + Q \cdot \xi_{-1,p,d} \qquad \forall p \in P,\, d \in D \tag{19}$$

$$a_d \leq t_p + u_{p,d} + A^{\max} \cdot (1 - \gamma_{p,d}) \qquad \forall p \in P,\, d \in D \tag{20}$$

$$a_d \geq R^{\text{hollow}} + y_d + \frac{1}{2} \cdot Q\,, \qquad b_d = y_d + z \qquad \forall d \in D \tag{21}$$

$$c \geq \sum_{d \in D} u_{p,d} + \frac{1}{2} \cdot Q \cdot \Big(\sum_{d \in D} \gamma_{p,d} - 2\delta_{-1,p} \Big) \qquad \forall p \in P \tag{22}$$

$$c \leq \sum_{d \in D} u_{p,d} + C^{\max} \cdot (1 - \delta_{-1,p}) \qquad \forall p \in P \tag{23}$$

$$h \geq c + \sum_{d \in D} v_d\,, \quad h \geq 2 \cdot z \tag{24}$$

The objective function, cf. Eq. (1), finds an optimal trade-off between three criteria: (i) the height of the transmission, (ii) the number of sliding sleeves, and (iii) the number of gear wheels. Since the exact height of the gearbox is difficult to express, we approximate it with the pseudo-height h, see Eq. (24). We use weighing parameters $W_1 \gg 1$ and $W_2 \ll 1$, i.e., we tolerate a larger gearbox in order to save sliding sleeves, but not to save gear wheels on the input shaft.

The overall transmission ratio for each gear g is an empirical value and given by Eq. (2). In hope of a more compact design, small deviations of 5% from the empirical reference values are granted, leading to an interval $[K_g^{\min}, K_g^{\max}]$. The transmission ratio is set by changing the wheel radii. To avoid huge wheels and a large axial expansion, two countershafts are introduced. The overall transmission is given as product of the pre-transmission (between input shaft and one countershaft), and the post-transmission (between the countershaft and the output shaft). The sum over the binary indicators $\xi_{g,p,d}$ identifies the matching post-transmission j_d for the pre-transmission i_g. Equations (3) and (4) describe the dependence of the pre- and post-transmission ratios from the gear wheel radii. To model the negative pre-transmission ratio i_{-1} of the reverse gear we use a sign function. For the reverse gear, an intermediate wheel on one of the drive shafts engages with a gear wheel on the input shaft and the reverse gear wheel on the other drive shaft. Fortunately, we do not need to include the intermediate wheel's radius s_{int}, since $i_{-1} = (s_{-1}/s_{\text{int}}) \cdot (s_{\text{int}}/r_{-1})$.

Equations (5)–(8) specify relations between gear assignment indicators. $\xi_{g,p,d}$ is active (equal to one) iff gear g is realized at the position given by plane p and drive shaft d. Variables $\gamma_{p,d}$, $\delta_{g,p}$ and ζ_p indicate if any $\xi_{g,p,d}$ is active over the respective set. Equation (6) ensures that each gear g is realized exactly once. Equation (9) make sure that (i) at most one component may be placed on a position (left), (ii) that each gear can be synchronized by a neighboring sleeve (middle), and (iii) that if the reverse gear wheel is placed at a certain position, another gear wheel is placed on the other drive shaft in the same plane (right). The input shaft is divided into a full shaft and a hollow shaft. ϕ_p indicates at which plane p the transition from full to hollow takes place. Equation (10) ensures that there is exactly one transition. Odd numbered gears are placed on full shaft planes, and even numbered gears are placed on hollow shaft planes, cf. Eq. (11). The reverse gear is placed on the hollow shaft, cf. Eq. (12). Equation (13) identifies the gear radii from the point of view of the gears (r_g, s_g) with the radii from the point of view of the gearbox positions $(t_p, u_{p,d})$. Equations (14) and (15) set the radius to the input shaft radius, if no gear wheel is realized on plane p.

Equations (16) and (17) determine the drive shaft radii $u_{p,d}$: If a synchronizer is realized, i.e., $\sigma_{p,d} = 1$, the radius equals the sleeve radius S^{sync}, if not, the radius equals the radius S^{min} of the drive shaft. The maximum radius v_d of wheels on drive shaft d in Eq. (18) is needed to compute the pseudo-height h.

Equations (19)–(20) determine the distance a_d between input shaft and drive shaft d. If a gear wheel is realized on drive shaft d in plane p, a_d is given by the sum of engaging gear radii, t_p on the input shaft, and $u_{p,d}$ on the drive shaft. If multiple gears are realized in different planes p, all have to agree on the same distance a_d. If no gear is realized, the distance between the two shafts has to be at least half of the tooth height Q. For the reverse gear, an extra space of at least the tooth height is needed. Equation (21) (left) makes sure that the distance a_d is large enough to separate the gear wheel on the final drive with radius y_d from the hollow input shaft. The distance b_d between drive shaft d and output shaft depends on the radii of the final drive, y_d and z, cf. Equation (21) (right). Between the two shafts, no gear wheels are allowed to engage except for the reverse gear wheel. The pseudo-height h is given by the maximum of (i) the drive shaft distance c plus the respective maximum radii on both shafts and (ii) the diameter of the gear wheel on the output shaft.

4 Results and Conclusion

We implemented the MINLP in the mathematical modeling framework JuMP [3]. For solving our instances, we chose the solvers Couenne [4] (version 0.5.6) and SCIP [5] (version 3.2.1). Both solvers guarantee global optimality for convex and nonconvex MINLPs. We varied our problem size by changing the number of gears and the number of possible axial planes. Table 2 shows the corresponding computing times. While the runtime increases with problem size, the runtimes are still very manageable. SCIP clearly outperforms Couenne for each instance.

Table 2 Runtimes on an Intel Core i5 with 2.4 GHz. Fewer planes than listed lead to infeasibility

No. of gears (incl. 1 reverse gear)	No. of available planes	Runtime in s	
		Couenne	SCIP
4 + 1	5	13	5
	6	21	9
5 + 1	6	28	5
	7	73	11
6 + 1	6	35	21
	7	168	37
7 + 1	7	436	117
	8	803	116

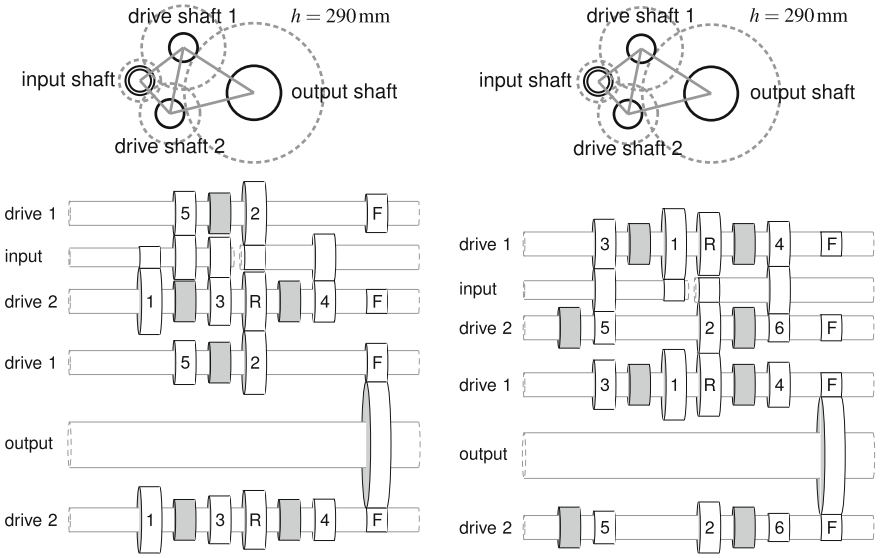


Fig. 1 Gearbox topology for 5 + 1 gears (*left*) and 6 + 1 gears (*right*). The *upper diagrams* depict the projection of the gearboxes in axial direction. The *lower diagrams* show a cut of the wheel engagements. A maximum of 7 planes (excluding the final drive) can be used

Figure 1 depicts exemplary solutions for the case of 5 + 1 gears and 6 + 1 gears and a maximum of 7 available axial planes (for further solutions with different numbers of gears and planes see [6]). Note that the gearboxes with 5 + 1 gears have a *larger* vertical expansion, i.e., height h, than the gearboxes with 6 + 1 gears. This is intended due to the chosen weighing of the objective criteria: 5 + 1 gears can be realized using only 3 sliding sleeves which is preferred to using a fourth sleeve that would have allowed for a more efficient packing in vertical direction.

Both MINLP solvers (Couenne and SCIP) find optimal topology proposals in the range of seconds to minutes. Having made a first step towards a gearbox design framework in this paper, we plan to integrate gear teeth design and shaft bending into our model in future work.

Acknowledgements The authors thank the German Research Foundation DFG for funding this research within the Collaborative Research Center SFB 805 "Control of Uncertainties in Load-Carrying Structures in Mechanical Engineering".

References

1. Fischer, R., Jrgens, G., Kkay, F., Najork, R., Pollak, B.: Das Getriebebuch. Springer (2012)
2. Salomon, S., Avigad, G., Purshouse, R.C., Fleming, P.J.: Gearbox design for uncertain load requirements using active robust optimization. Eng. Optim. **48**(4), 652–671 (2016)
3. Lubin, M., Dunning, I.: Computing in operations research using Julia. INFORMS J. Comput. **27**(2), 238–248 (2015)
4. Belotti, P., et al.: Branching and bounds tightening techniques for non-convex MINLP. Optim. Meth. Softw. **24**(4–5), 597–634 (2009)
5. Achterberg, T.: SCIP: solving constraint integer programs. Math. Program. Comput. **1**(1), 1–41 (2009)
6. Dörig, B., et al.: Gearbox design via mixed-integer programming. In: Proceedings of the VII European Congress on Computational Methods in Applied Sciences and Engineering (2016)

Line Planning on Path Networks with Application to the Istanbul Metrobüs

Ralf Borndörfer, Oytun Arslan, Ziena Elijazyfer, Hakan Güler, Malte Renken, Güvenç Şahin and Thomas Schlechte

Abstract Bus rapid transit systems in developing and newly industrialized countries often consist of a trunk with a path topology. On this trunk, several overlapping lines are operated which provide direct connections. The demand varies heavily over the day, with morning and afternoon peaks typically in reverse directions. We propose an integer programming model for this problem, derive a structural property of line plans in the static (or single period) "unimodal demand" case, and consider approaches to the solution of the multi-period version that rely on clustering the demand into peak and off-peak service periods. An application to the Metrobüs system of Istanbul is discussed.

R. Borndörfer (✉) · Z. Elijazyfer · M. Renken · T. Schlechte
Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
e-mail: borndoerfer@zib.de

Z. Elijazyfer
e-mail: elijazyfer@zib.de

M. Renken
e-mail: renken@zib.de

T. Schlechte
e-mail: schlechte@zib.de

O. Arslan
IVU Traffic Technologies AG, Borchersstr. 20, 52072 Aachen, Germany
e-mail: oar@ivu.de

H. Güler
Engineering Faculty, Transportation Engineering Division, Civil Engineering Department, Sakarya University, Esentepe, 54187 Sakarya, Turkey
e-mail: hguler@sakarya.edu.tr

G. Şahin
Industrial Engineering, Sabanci University, Orhanli, 34956 Tuzla, Istanbul, Turkey
e-mail: guvencs@sabanciuniv.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_32

1 Introduction

The establishment of a public transportation system involves decision making on *network design*, *line planning*, *timetabling*, and *fare planning*. Due to the (contradictory) objectives of *minimizing the costs* and *maximizing the level of service*, each task is already challenging on its own, such that the planning process is typically conducted in succession [1]. Its integrated treatment has been taken up only recently [2], and it is still quite unclear how a "globally optimal system" should look like or how it could be identified. The investigation of basic, but practically relevant, classes of networks is one way to advance in this direction. We consider an interesting type of transportation system with the simplest possible network structure: a *path topology* is often found in *bus rapid transit* (BRT) systems in developing or newly industrialized countries, such as Trolébus in Quito [5], or our subject of investigation, the Metrobüs in Istanbul, see Fig. 1. Even though such systems may, at first sight, look small in terms of numbers of stations or links, they typically service the bulk of the demand, exerting a trunk function.

In BRT systems such as Metrobüs, the demand is typically highly asymmetric w.r.t. its distribution on the line and notably fluctuating w.r.t. time: There is a morning peak towards the center, an evening peak towards the outskirts, and significantly less demand during other times, see Fig. 2. Traditional line planning addresses this demand fluctuation by constructing a base service, which is augmented in peak hours, or vice versa. But is this prevalent procedure conclusive? This is the question that we study in this article. In fact, as far as we know, *multiperiod line planning* (much less time continuous) has not been considered in the optimization literature so far.

We study the line planning problem on path networks over a planning horizon of an entire day. In a path network, the passenger *travel paths* are uniquely determined.

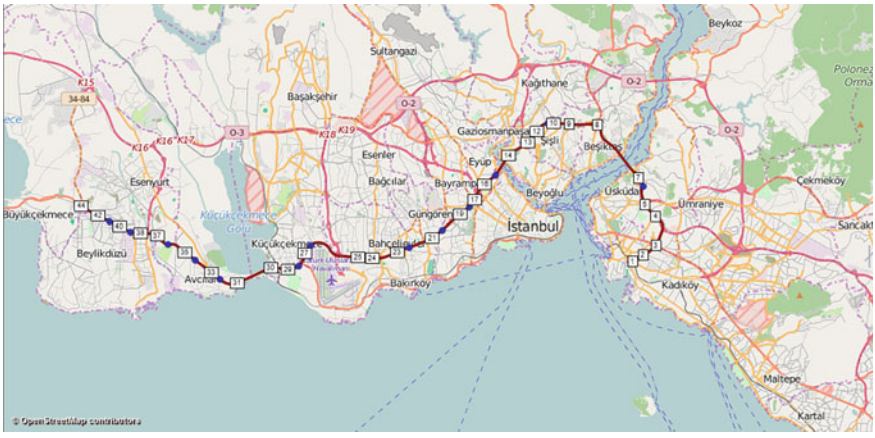
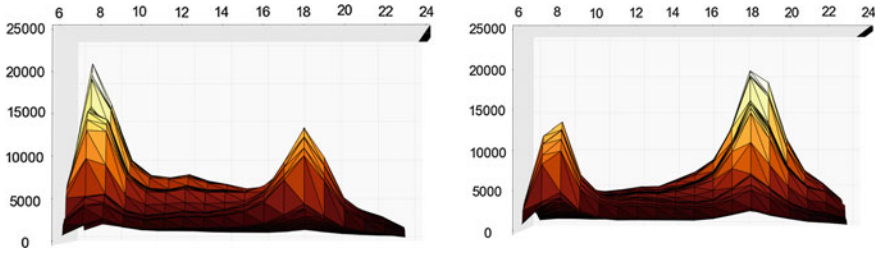


Fig. 1 The Metrobüs in Istanbul with its 44 stations (visualization by `PTV Visum` [4])



(a) Demands in direction from east to west. (b) Demands in direction from west to east.

Fig. 2 Traffic demand in the Istanbul Metrobüs system over 24 h, summed over all OD pairs

This gives rise to a fixed passenger *load* (or volume) on every individual link of the path network. These loads have to be covered by lines that must offer appropriate capacities and frequencies. We concentrate on the two dominating features of the demand characteristics of the problem: (i) asymmetricity of the demand towards the center of the network creates a *unimodal (increasing to a maximum and then decreasing) demand distribution* along the path, (ii) the level of difference in the total number of passengers between peak and off-peak hours requires *partitioning the day into multiple sections*. We show that unimodal demand leads to a unimodal line plan in the single period case. This gives an indication that line plans that augment a base service in peak hours might work well, i.e., that augmentations in space and time are reasonable. A computational comparison of two solution approaches for the multiperiod case on Metrobüs data corroborates this claim.

2 The Static Demand Case

We study the following *demand coverage model* to assign frequencies to a given set of lines, that are modeled as paths on a traffic network, which in our setting itself has path topology, in the static demand or single period case. Consider a *traffic network* $N = (V = [n], A = A^{\rightarrow} \cup A^{\leftarrow})$ as a path with *forward arcs* $A^{\rightarrow} = \{(i, i+1) : i \in [n-1]\}$ and *backward arcs* $A^{\leftarrow} = \{(i+1, i) : i \in [n-1]\}$. Let $c : A \to \mathbb{R}^+$ be a *cost* and $d : A \to \mathbb{R}_0^+$ the *demand*, i.e., the number of passengers traveling across each arc. Let the set of *lines* L consist of all non-trivial (directed) paths on N with endpoints in some set $T = \{t_1, \ldots, t_m\} \subseteq V$ of *terminal stations*, $t_1 < \cdots < t_m$, and denote the beginning and end of line $l \in L$ by $\alpha(l)$ and $\omega(l)$. It is reasonable to require the beginning and end of N to be in T, i.e., $t_1 = 1$ and $t_m = n$. A function $p : L \to \mathbb{N}_0$ is called a *line plan*, and $p(l)$ is the frequency of line l. We say that a line plan is *balanced* if the flow balance conditions $\sum_{l \in L : \alpha(l) = v} p(l) = \sum_{l \in L : \omega(l) = v} p(l)$ hold for every vertex v. The *(capacity) supply* provided by p is $s : A \to \mathbb{R}_0^+, a \mapsto \sum_{l \in L, a \in l} p(l)\kappa$, where κ is some positive constant specifying the number of passengers that can be transported per vehicle. p is *feasible* (for the demand d) if it is balanced and

if $s(a) \geq d(a)$ for every $a \in A$. A feasible line plan p is *optimal* if it minimizes the *overall cost* $C := \sum_{l \in L: p(l)>0} c^f(l) + \sum_{l \in L} c^o_l p(l)$, where $c^f : L \to \mathbb{R}^+_0$ denotes a fixed setup cost for each line, and $c^o : L \to \mathbb{R}^+_0, l \mapsto \sum_{a \in l, a \in A} c(a)$ the operational cost per vehicle.

The demand coverage model can be formulated as an integer program as follows. Let $x_l \in \{0, 1\}$ be a binary variable that takes value one if line $l \in L$ is selected, and v_l an integer variable that defines how many vehicles are operating line l.

$$\min \sum_{l \in L} c^f_l x_l + \sum_{l \in L} c^o_l v_l \qquad (\mathrm{DCM}_{\mathrm{MIP}})$$

$$\sum_{l \in L_a} \kappa v_l \geq d_a \qquad \forall a \in A, \qquad (1)$$

$$M x_l - v_l \geq 0 \qquad \forall l \in L, \qquad (2)$$

$$\sum_{l \in L} v_l \leq V \qquad (3)$$

$$x_l \in \{0, 1\} \qquad \forall l \in L, \qquad (4)$$

$$v_l \in \mathbb{N} \qquad \forall l \in L. \qquad (5)$$

Inequalities (1) of program ($\mathrm{DCM}_{\mathrm{MIP}}$) make sure that the demand d_a on each segment $a \in A$ is covered by the capacity of the set L_a of lines that contain a. The v-variables are coupled with the x-variables via inequalities (2) in order to assure that only chosen lines are assigned a positive number of up to M vehicles each. Constraint (3) limits the overall number of vehicles. The objective of program ($\mathrm{DCM}_{\mathrm{MIP}}$) models both the minimization of fixed and operational costs.

3 Unimodality

By examination of the passenger data for the Istanbul Metrobüs system it becomes apparent that, for any given time, the distribution of the passenger demand volume per station is approximately unimodal: More centrally located stations experience higher traffic. Under these conditions, and if the fixed costs c^f are negligible, an optimal line plan is unimodal as well, i.e., for any two lines, one is contained in the other. This greatly reduces the complexity of the problem and allows to solve it in linear time.

Denote by $\bar{a}$ the reverse of arc $a \in A$, and by $d^{\leftrightarrow}(a) := \max\{d(a), d(\bar{a}))\}$ the maximum demand on $a \in A$ in both directions. A line plan p is *unimodal* if $p(l)p(l') = 0$ for any two forward lines l and l' that do not satisfy $[\alpha(l), \omega(l)] \supseteq [\alpha(l'), \omega(l')]$ or $[\alpha(l'), \omega(l')] \supseteq [\alpha(l), \omega(l)]$, and similar for backward lines.

Theorem 1 *If the maximum demand is unimodal and the fixed costs are zero, then there is an optimal unimodal line plan.*

Proof It is easy to see that every balanced line plan p satisfies $s(a) = s(\bar{a})$ for all $a \in A$. Therefore, p is feasible if and only if $s(a) \geq d^{\leftrightarrow}(a)$ for all $a \in A^{\rightarrow}$, i.e., we can restrict our attention to $A^{\rightarrow}$.

Call two forward arcs equivalent if they are not separated by a terminal. Then any two equivalent arcs $a \sim a'$ are always assigned the same capacity supply by every line plan, because they are covered by the same set of lines. For any forward arc $a \in A^{\rightarrow}$, let

$$s_*(a) := \max_{a' \sim a} d^{\leftrightarrow}(a') \qquad \text{and} \qquad k_*(a) := \lceil s_*(a)/\kappa \rceil.$$

Then any feasible line plan must provide a capacity supply of at least $s_*(a)$ on $a \in A^{\rightarrow}$, and use at least $k_*(a)$ vehicles. The unimodality of $d^{\leftrightarrow}$ implies unimodality of s_* and, in turn, of k_*.

It is easy to construct a unimodal line plan p on $A^{\rightarrow}$ with supply $s = s_*$, using exactly $k_*(a)$ vehicles on any arc. Namely, let $l_{1,n}$ be the line from $t_1 = 1$ to node $t_m = n$, set

$$p_*(l_{1,n}) := \min\{k_*(1,2), k_*(n-1,n)\},$$

subtract $p_*(l_{1,n})$ from $k_*(a)$ for all $a \in A^{\rightarrow}$, and remove all arcs such that $k_*(a) = 0$; then iterate until all arcs have been removed. Proceed in the same way for $A^{\leftarrow}$.

The resulting line plan is unimodal, feasible, and also optimal, because its cost

$$C = \sum_{l \in L} p(l) \sum_{\substack{a \in l \\ a \in A^{\rightarrow}}} c(a) = \sum_{a \in A^{\rightarrow}} \frac{s(a)}{\kappa} = \sum_{a \in A^{\rightarrow}} \frac{s_*(a)}{\kappa} = \sum_{a \in A^{\rightarrow}} k_*(a)$$

is minimal by minimality of k_*. □

4 The Multiperiod Case

The case at hand, the Istanbul Metrobüs system, is characterized by a highly fluctuating and asymmetrical demand caused primarily by commuters. In the morning hours a high volume of passengers needs to be transported from the periphery to the center, while the flow is reversed in the evening hours. In between the demand falls sharply. Figure 2 gives a more detailed impression of the passenger demands in both directions. In the figure, the x-axis accounts for time and the y-axis for the corresponding demand. The peak-times stand out, and there is a relatively constant demand in between. At any point in time, the spatial demand distribution is very close to being unimodal.

Fig. 3 Identifying peak and off-peak times by clustering hourly line plans

We identify the peak and the off-peak time intervals as follows. Running the demand covering model (DCM_{MIP}) for every hour from 6 to 23 o'clock produces 18 line plans, represented by their p-vectors. We check these vectors for similarities using the classical k-means clustering algorithm [3]. Figure 3 shows the results for $k = 2$: The algorithm clusters the time windows from 7 to 9 and 17 to 20 together, which correspond to the peaks. All other time windows form a common off-peak section.

Two line planning strategies that exploit this temporal subdivision suggest themselves: On the one hand it might be advisable to run separate schedules during peak and off-peak times to cope with the different demands. On the other hand, the peak demands exceed the base demands. It might therefore be possible to run a continued base line plan throughout the entire day, augmented by additional resources during peak times. Such an augmentation can be seen as an analogon over time of a unimodal line plan, which can be seen as a capacity augmentation in space. Our theoretical findings on unimodal line plans suggest that resorting to unimodal line plans is feasible for almost unimodal demands. In combination with a similar result over time, this would make a good case for this traditional planning procedure.

We consider the following two computational scenarios in a setting without fixed costs and operational costs based on driven distance:

(1) a discontinued base schedule with independent peak time line plans and
(2) a continued base schedule with peak time line plans on top.

Computations were performed on a Intel i7-4790 3.60 GHz CPU using CPLEX 12.6 as a MIP solver (Table 1).

The computations corroborate our expectations that service augmentation in space and time produces results whose quality is on a par with unrestricted planning w.r.t. operational costs.

Table 1 Comparing different line planning approaches for the Istanbul Metrobüs on weekdays

Scenario	MIP solution		Unimodular solution	
	Value (km)	Computation time (s)	Value (km)	Computation time (s)
(1)	12 041	1.9	12 056	0.9
(2)	12 144	1.8	12 164	0.8

References

1. Borndörfer, R., Grötschel, M., Jaeger, U.: Planning problems in public transit. Production Factor Mathematics, pp. 95–122 (2010). doi:10.1007/978-3-642-11248-5
2. Liebchen, C.: Linien-, Fahrplan-, Umlauf- und Dienstplanoptimierung: Wie weit können diese bereits integriert werden? In: Heureka'08. Stiftung Heureka, FGSV Verlag (2008)
3. McCallum, A., Nigam, K., Ungar, L.H.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of ACM, KDD '00, pp. 169–178. ACM, New York, NY, USA (2000). doi:10.1145/347090.347123
4. PTV AG. http://vision-traffic.ptvgroup.com/en-us/products/ptv-visum/
5. Torres, L.M., Torres, R., Borndörfer, R., Pfetsch, M.E.: Line Planning on paths and tree networks with applications to the quito trolebus system. In: Fischetti, M., Widmayer, P. (eds.) ATMOS'08, OpenAccess Series in Informatics (OASIcs), vol. 9. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany (2008). doi:10.4230/OASIcs.ATMOS.2008.1583. http://drops.dagstuhl.de/opus/volltexte/2008/1583

Particle-Image Velocimetry and the Assignment Problem

Franz-Friedrich Butz, Armin Fügenschuh, Jens Nikolas Wood and Michael Breuer

Abstract The Particle-Image Velocimetry (PIV) is a standard optical contactless measurement technique to determine the velocity field of a fluid flow for example around an obstacle such as an airplane wing. Tiny density neutral and light-reflecting particles are added to the otherwise invisible fluid flow. Then two consecutive images (A and B) of a thin laser illuminated light sheet are taken by a CCD camera with a time-lag of a few milliseconds. From these two images one tries to estimate the local shift of the particles, for which it is common to use a cross-correlation function. Based on the displacement of the tracers and the time-lag, the local velocities can be determined. This method requires a high level of experience by its user, fine tuning of several parameters, and multiple pre- and post-processing steps of the data in order to obtain meaningful results. We present a new approach that is based on the matching problem in bipartite graphs. Ideally, each particle in image A is assigned to exactly one particle in image B, and in an optimal assignment, the sum of shift distances of all particles in A to particles in B is minimal. However, the real-world situation is far from being ideal, because of inhomogeneous particle sizes and shapes, inadequate illumination of the images, or particle losses due to a divergence out of the two-dimensional light sheet area into the surrounding three-dimensional space, to name just a few sources of imperfection. Our new method is implemented in MATLAB with a graphical user interface. We evaluate and compare it with the cross-correlation method using real measured data. We demonstrate that our new method requires less interaction with the user, no further post-processing steps, and produces less erroneous results. This article is based on the master thesis [5], written by the first coauthor, and supervised by all other coauthors.

F.-F. Butz (✉) · A. Fügenschuh · J.N. Wood · M. Breuer
Helmut-Schmidt-University/University of the Federal Armed Forces Hamburg,
Holstenhofweg 85, 22043 Hamburg, Germany
e-mail: butz@hsu-hh.de

A. Fügenschuh
e-mail: fuegenschuh@hsu-hh.de

J.N. Wood
e-mail: wood@hsu-hh.de

M. Breuer
e-mail: breuer@hsu-hh.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_33

1 Short Introduction into PIV

Many fields of modern science and engineering include applications that require information about fluid flow phenomena. The examples range from medical investigations such as the blood flow inside the human body to the optimization of the aerodynamic behavior of wind turbine blades.

In many cases these flow fields are determined experimentally. The majority of real-life flow phenomena involve highly complex characteristics. Due to this, the measurement equipment must be able to predict the flow within the field of interest.

In modern science contactless flow measurements have become a standard. These techniques offer the advantage of non-invasive data aquisition since the flow is not disturbed by any intrusive probe. A common and very widely used application is particle-image velocimetry (PIV), where the flow field is illuminated by a strong laser. An optical lense is used to transfer the laser beam into a light sheet, which makes it possible to measure a two-dimensional flow field. For this purpose, the flow is seeded with small particles (water flow) or droplets (air flow) also called tracers. During this study silver-coated hollow glass spheres (S-HGS, see Fig. 1a) of small diameter (around 20 μm) are used in a water flow [1, 6, 7]. The tracers are assumed to be density-neural and therefore follow the flow with minimum interference.

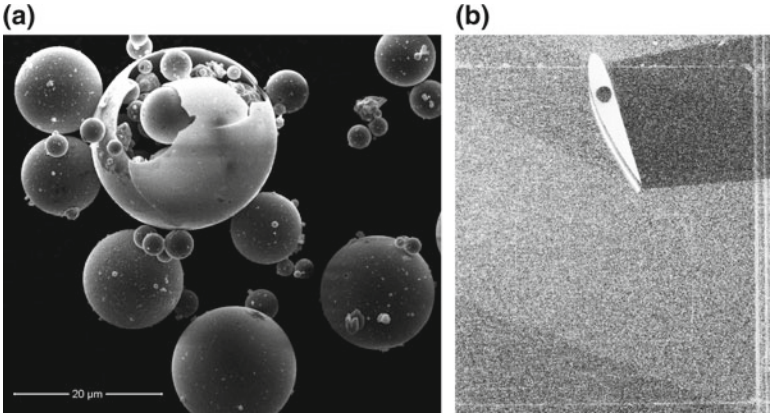


Fig. 1 **a** S-HGS particles, manufactured by Dantec Dynamics. Image taken with a scanning electron microscope of the Institute of Materials Technology, Helmut-Schmidt-University. **b** PIV image of a wing cross-section. Total area approx. 20×20 cm. Laser beam is coming from the *left*. For a better visibility, contrast and brightness were adjusted

2 Set-Up of PIV Experiments

In the present investigation the fluid-particle flow in a water tunnel is photographed using a high-resolution CCD digital camera [2]. The camera's optic is focused on a thin layer of the flow that is enlightened by a short laser pulse light beam. The silver of the coated hollow glass spheres within that layer reflects the light to a high degree, whereas other particles in front of and behind that layer remain in the shadow and out of the focus, see Fig. 1b. In this way, two images are taken consecutively within a short time lag Δt. The images in this study typically show 50,000–200,000 PIV tracers. One has to deal with all kinds of technical hurdles of a typical imperfect measurement, such as outliers, pixel errors, or shaded regions, which can be seen in Fig. 1b.

3 Classical Cross-Correlation Method

Having two digital PIV images (called A and B) at hand, one tries to determine the direction of the flow at each coordinate (x, y) of the image. To this end, the image is split up into rectangular segments or tiles (for example, 32×32 or 64×64 pixels each). In the classical method, the brightness value (an integer between 0 and 4095) defines the particle density for each pixel of the image. An image thus can be considered as a two-dimensional density function $A(x, y) \in \{0, \ldots, 4095\}$ for each (x, y) of the image (having a typical resolution of 2048×2048). The average movement direction for each tile is then determined by computing the average movement based on this density.

A mathematical algorithm for this task is the cross-correlation function (also known as sliding inner-product). It is a measure for the similarity of two time series or functions (here: A and B), and it is (in our case) a two-dimensional function in (ξ, η), which is the lag (or shift vector) of the function values in A to those of B. In our discrete setting, the cross-correlation is defined as

$$\Phi_{A\star B}(\xi, \eta) := \sum_{i=0}^{M} \sum_{j=0}^{N} A(i,j) \cdot B(i+\xi, j+\eta),$$

where M, N is the size of the tile. For the input shown in Fig. 2 (A and B), the corresponding cross-correlation function $\Phi_{A\star B}$ is shown on the right of this figure. Note that Φ has a maximum, which corresponds to the linear shift of the density of A to match the density of B in the best way. The corresponding vector (ξ^*, η^*) which is the argument of this maximum is taken as the local movement vector for this particular tile. Since Φ is a discrete function having no analytical properties, this maximum is computed by testing each potential value of (ξ, η) within a certain range. The result of this method is shown in Fig. 3a. In this test case, the flow around an airfoil with an incoming velocity of 2.8 m/s is considered flowing from the top to the bottom of the

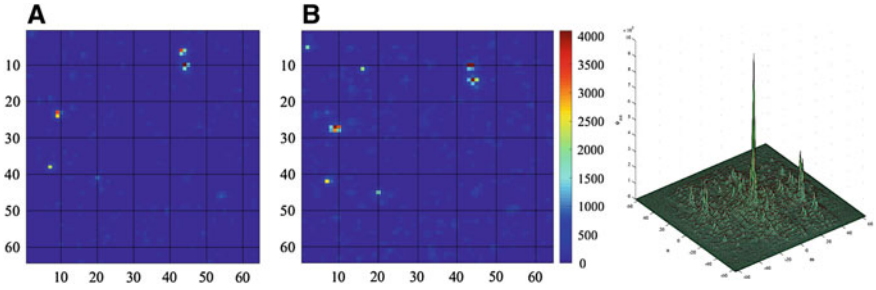


Fig. 2 Image pair *A*, *B*, and the corresponding cross-correlation function

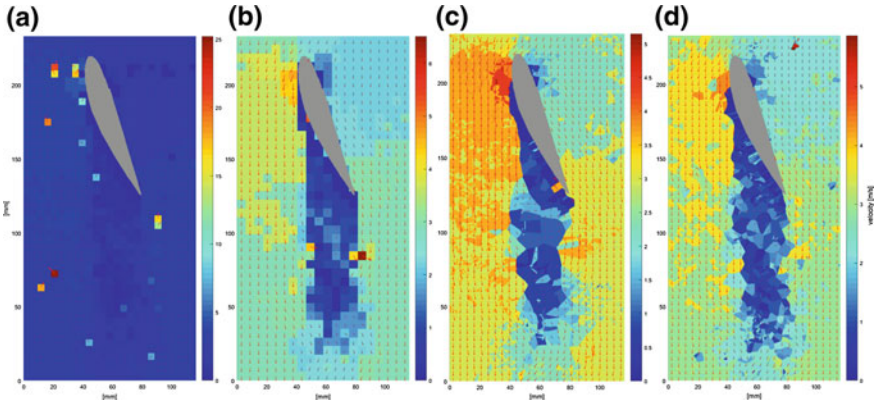


Fig. 3 Flow around a wing cross-section with an incoming velocity of 2.86 m/s. From *left* to *right*: **a** cross-correlation method, **b** most common vector method, **c** assignment method with counting objective function, **d** assignment method with similarity objective function

water tunnel. One can clearly see a large recirculation area at the leeward side of the airfoil. However, there are also outliers visible, where the cross-correlation method assumed velocities of around 25 m/s in an opposite direction to the surrounding flow field, which is technically impossible and therefore meaningless.

4 New Analytical Methods

We describe three new methods to compute a two-dimensional flow vector field based on the experimental data, i.e., two PIV-images taken within a short time lag. All three methods require as a first step to detect the individual particles in each tile of the two images, and store them in two lists $\alpha = (a_1, \ldots, a_m)$ and $\beta = (b_1, \ldots, b_n)$ [4]. (Note that α and β also depend on the tile, but for notational simplicity we neglect the index of the tile.) We emphasize that the number of particles in these two lists are not necessarily equal.

4.1 Most Common Vector

The idea of the first method is to identify a vector Δ^* that (ideally) maps the list of points α onto list of points β, that is, in set notation: $\beta = \alpha + \Delta^*$. Because of potential outliers, measurement errors, and misidentification of particles in both images, this equality holds only in an approximative sense. In order to identify Δ^*, we consider all vectors $\Delta_{i,j} := b_j - a_i$. This gives a matrix of transition vectors $D = (\Delta_{i,j})_{i,j}$, in which we count the number of occurrences of each vector $\Delta_{i,j}$. The most common vector (i.e., the vector that occurs most often in D) is the transition vector Δ^*.[1] The result of this method is shown in Fig. 3b. In comparison to the classical cross-correlation method, the number and the strength of outliers is significantly reduced. Both methods still have in common that only one transition vector is computed per tile.

4.2 Assignment Method

The assignment problem (also known as transportation problem with 0/1 supply and demand, or weighted matching problem in complete bipartite graphs [3]) can be formulated as the following integer programming problem. We introduce integer decision variables $x_{i,j} \in \{0, 1\}$, if particle a_i in list α is assigned to (moved to) particle b_j in list β. Assuming w.l.o.g. that the number of particles in β is greater or equal than the number of particles in α, each particle in α is assigned to exactly one particle in β, and each particle in β is assigned to at most one particle in α:

$$\forall\, i \in \alpha \,:\; \sum_{j \in \beta} x_{i,j} = 1,$$
$$\forall\, j \in \beta \,:\; \sum_{i \in \alpha} x_{i,j} \leq 1.$$

The objective is to maximize a preference measure that takes into account how well particle i would fit to particle j when being assigned:

$$\max \sum_{i \in \alpha} \sum_{j \in \beta} c_{i,j}\, x_{i,j}.$$

We developed two different methods how to come up with practically meaningful values for $c_{i,j}$.

[1] This statement was formalized and rigorously proven by Fabian Gnegel at Helmut-Schmidt-University Hamburg.

4.2.1 Objective Function I: Counting

The first method is similar to the "most common vector" method. For each pair of particles (a_i, b_j) with distance vector $\Delta_{i,j}$ (as above), we set $c_{i,j}$ to the number of $\Delta_{i,j}$ in the submatrix $(\Delta_{k,l})_{k,l}$ with $k \in \alpha \backslash \{i\}$ and $l \in \beta \backslash \{j\}$. With the so-defined $c_{i,j}$, we solve a maximum-weight assignment problem. The improved results are depicted in Fig. 3c.

4.2.2 Objective Function II: Similarity

In the second method, we compute for each pair of particles (a_i, b_j) the cosine of the angle between $\Delta_{i,j}$ and $\Delta_{k,l}$ for all $k \in \alpha \backslash \{i\}$ and $l \in \beta \backslash \{j\}$. Each cosine is a real number between -1 and 1, where 1 means that $\Delta_{i,j}$ and $\Delta_{k,l}$ are pointing into the same direction, 0 means they are perpendicular, and -1 means they are pointing in the opposite direction. Summing up all these cosine values then yields $c_{i,j}$. With the so-defined $c_{i,j}$, we again solve a maximum-weight assignment problem. Figure 3d shows the corresponding result.

5 Conclusions

The outcomes of the assignment methods shown in Fig. 3c and d are advantageous compared to the previous results. Due to the assignment it is possible to have an individual transition information for each detected particle (not just one per tile). Qualitatively, our new methods have a higher "resolution" compared to the classical cross-correlation method. The next step of the research aims at a suitable measure that is able to describe the quantitative superiority of one method over the other.

References

1. Adrian, R.J.: Particle-imaging techniques for experimental fluid mechanics. Annu. Rev. Fluid Mech. **23**(1), 261–304 (1991)
2. Barbe, D.F.: Imaging devices using the charge-coupled concept. Proc. IEEE **63**(1), 38–67 (1975)
3. Burkard, R., Dell'Amico, M., Martello, S.: Society for industrial and applied mathematics. In: Assignment Problems. Philadelphia (2012)
4. Butz, F.-F.: Entwicklung und Implementierung eines Algorithmus zur Detektion von Streuteilchen in PIV-Aufnahmen. Angewandte Mathematik und Optimierung, Schriftenreihe AMOS#40. Helmut-Schmidt-University, Hamburg (2016)
5. Butz, F.-F.: Entwicklung und Implementierung von Analysemethoden zum Erfassen von Geschwindigkeitsfeldern mit dem PIV-Verfahren. Angewandte Mathematik und Optimierung, Schriftenreihe AMOS#45. Helmut-Schmidt-University, Hamburg (2016)

6. Raffel, M., Willert, C.E., Kompenhans, J.: Particle Image Velocimetry: A Practical Guide. Engineering online library. Springer, Berlin (1998)
7. Westerweel, J.: Fundamentals of digital particle image velocimetry. Meas. Sci. Tech. **8**(12), 1379–1392 (1997)

Analysis of Operating Modes of Complex Compressor Stations

Benjamin Hiller, René Saitenmacher and Tom Walther

Abstract We consider the modeling of operation modes for complex compressor stations (i.e., ones with several in- or outlets) in gas networks. In particular, we propose a refined model that allows to precompute tighter relaxations for each operation mode. These relaxations may be used to strengthen the compressor station submodels in gas network optimization problems. We provide a procedure to obtain the refined model from the input data for the original model.

1 Introduction

Gas transmission networks are a crucial part of the European energy supply infrastructure. The gas flow is driven by pressure potentials. To maintain the necessary pressure levels and control the routing of the gas in the network, compressor stations are used. In the German network compressor stations usually interconnect two or more pipeline systems. They often have a complex internal structure, allowing them to realize different routing patterns between the boundary nodes, which may serve as inlet or outlet depending on the requirements of the surrounding network [2]. An example of such a complex compressor station is shown in Fig. 1.

In this paper, we consider the compressor station modeling introduced in [2]. This model combines a network containing compressors and valves and a set of switching states for these elements to describe all feasible operation modes of a compressor

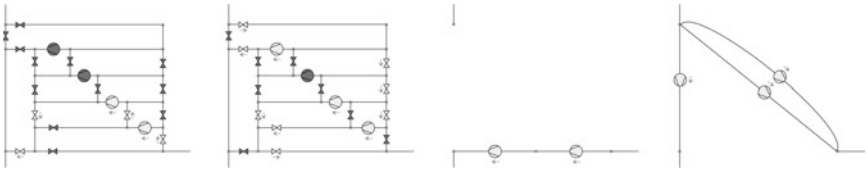
The authors thank the DFG for their support within project A04 in CRC TRR154 and the BMBF Research Campus Modal (fund number 05M14ZAM) and ICT COST Action TD1207 for additional support.

B. Hiller (✉)
Zuse Institute Berlin, Takustraße 7, 14195 Berlin, Germany
e-mail: hiller@zib.de

R. Saitenmacher
e-mail: saitenmacher@zib.de

T. Walther
e-mail: walther@zib.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_34



(a) Flow from east to south, double serial compression. Elements colored dark are "closed".

(b) Flow from north to east with double and to south with single compression.

(c) Reduced representation of operation mode in (a).

(d) Reduced representation of operation mode in (b).

Fig. 1 Two operation modes of a large compressor station (**a, b**) and their reduced representations (**c, d**) obtained with the methods of Sect. 3

station. The constraints describing the technical capability of a compressor may be nonlinear and nonconvex, leading to hard-to-solve MINLP models for a compressor station. To improve the model, a natural idea is to precompute, for each operation mode, bounds on the minimum and maximum flow and pressure that can be handled and to include this information in the model. This should help the solution process to detect unsuitable operation modes early. However, the modeling of an operation mode from [2] does not specify whether a compressor is actively compressing or bypassed. Thus, no nontrivial flow bounds may be obtained for an operation mode.

Contribution We develop techniques for analyzing the original representation of operation modes to obtain a more detailed representation prescribing for each compressor whether it is compressing or in bypass. This allows to compute tight bounds (or even convex hulls) for the pressure/flow combinations that can be handled by each operation mode. The crucial ingredient is a method to obtain a reduced representation of an operation mode to cope with redundancies due to the original representation. Examples of such reduced representations are also shown in Fig. 1.

Related work We briefly mention some related papers and refer to [5] for a comprehensive overview. Most work on optimization of compressor stations has focused on simple compressor stations compressing from a single inlet to a single outlet. The fact that a compressor station usually features several (often distinct) compressor units has been dealt with by using an aggregated model, like a range for the power required by the compression process [3], box constraints for flows and pressures [1], or a polyhedral model [6]. Papers using a detailed model for the operation of a single compressor unit usually assume that a compressor station consists of several parallel identical units [7], the only discrete decision being the number of units switched on. A recent exception is the work of [4], considering configurations consisting of serial stages of units used in parallel. Complex multi-way compressor stations with multiple operating modes are only considered in [2] and related work.

The remaining paper ist structured as follows. Section 2 recalls the model introduced in [2]. In Sect. 3, we propose a method that reduces the description of a single operation mode to a kind of "normal form". This is used in order to detect redundancy and generate a set of redundancy-free operation modes that, as a set, are equivalent to the original operation modes. Finally, we report on some computational results in Sect. 4.

2 Model for Complex Compressor Stations

Our model for compressor stations closely follows that proposed in [2]. We represent a compressor station as a directed graph $(V, A_{\text{va}} \cup A_{\text{cg}} \cup A_{\text{sc}})$, where the arc set consists of the set of valves A_{va}, the set of compressors A_{cg}, and the set of shortcuts A_{sc}. Moreover, we partition the set of nodes into boundary nodes $V_{\pm}$ and inner nodes V_0. For each node $u \in V$ we introduce a variable for the pressure p_u with non-negative lower and upper bounds $\underline{p}_u$ and $\overline{p}_u$. For each arc $a \in A$ there is a variable for the mass flow q_a with lower and upper bounds $\underline{q}_a$ and $\overline{q}_a$. Positive mass flow values indicate flow in the direction of the arc, whereas negative values represent flow in the opposite direction. The precise values of the bounds depend on the type and state of an element. We define the excess of mass flow at nodes by

$$b_u := \sum_{a \in \delta^-(u)} q_a - \sum_{a \in \delta^+(u)} q_a \quad \text{for all } u \in V. \tag{1}$$

where $\delta^-(u)$ and $\delta^+(u)$ denote the sets of ingoing and outgoing arcs for node u. At inner nodes, the mass flow is conserved, i.e., we have $b_u = 0$ for $u \in V_0$.

Valves can be *open* or *closed* and are used to control the route of gas through the compressor station. A binary variable s_a distinguishes between these states ($s_a = 1$: *open*, $s_a = 0$: *closed*). A closed valve is like a missing connection, i.e., there is no flow and the pressures are decoupled. An open valve admits arbitrary flow and the pressures at its nodes are identical.

Compressors may operate in one of the states *closed* (no gas flow), *active* (compressing), and *bypass* (gas flow without compression). Binary variables $s_a, s_a^{\text{ac}}, s_a^{\text{bp}}$ distinguish between the states *active*, *bypass* and *closed* where $s_a = 1, s_a^{\text{ac}} = 1$ corresponds to *active*, $s_a = 1, s_a^{\text{bp}} = 1$ corresponds to *bypass* and $s_a = 0$ corresponds to *closed*. A closed compressor again corresponds to a missing connection and one in bypass to an open valve. We model the capabilities of an active compressor $a \in A_{\text{cg}}$ by an abstract set $P_a \subseteq \mathbb{R}^3_{\geq 0}$ of feasible inlet pressure, outlet pressure, and mass flow. Thus our methods apply to a large range of compressor models. The constraints describing P_a, the capability set of a compressor, may be nonlinear and nonconvex, leading to hard-to-solve MINLPs for the entire compressor station.

Shortcuts are convenient modeling elements that allow arbitrary gas flow between two nodes without pressure drop.

An *operation mode* specifies the switching state of each active element (valves, compressors) and thus the route of the gas flow through the compressor station. Operation modes are modeled in [2, Section 6.1.8] by a triple $(A_{\text{active}}, \mathscr{M}, d)$, where $A_{\text{active}} = A_{\text{va}} \cup A_{\text{cg}}$ is the set of active elements. The set $\mathscr{M} \subseteq \{0, 1\}^{A_{\text{active}}}$ describes each operation mode $m \in \mathscr{M}$ by stating whether an active element a is *open* ($m_a = 1$) or *closed* ($m_a = 0$). In the case of an *open* compressor it is not yet specified whether this compressor is in *bypass* or is *active*. Finally, the function $d : A_{\text{active}} \times \mathscr{M} \to \{-1, 0, 1\}$ describes whether the flow direction for an active arc $a = (u, v)$ is

restricted or not (-1: flow in opposite direction of arc, 0: direction unspecified, -1: flow in arc direction).

As mentioned in the introduction, the fact that this representation does not specify whether an *open* compressor is *active* or running in *bypass* precludes us from obtaining tight bounds for flows and pressures obtainable by an operation mode. We thus propose a more detailed representation where each operation mode is *fully specified* by prescribing for each compressor whether it is *active* or in *bypass*. To obtain this representation from the original one we enumerate all *active/bypass* combinations for each operation mode. Since this leads to many and redundant operation modes, we apply the methods from Sect. 3 to obtain an equivalent smaller set of fully specified operation modes. These are described by a tuple $(A_{\text{active}}, \mathscr{M}^{\text{va}}, \mathscr{M}^{\text{cg}}, d')$, where $\mathscr{M}^{\text{va}} \subseteq \{0,1\}^{A_{\text{va}}}$ prescribes the state of each valve and $\mathscr{M}^{\text{cg}} \subseteq \{0,1\}^{A^2_{\text{cg}}}$ prescribes the state of each compressor. For each of these operation modes, we can now compute tight pressure and inflow bounds by solving the optimization problem given by (4)–(8) together with respective objective functions. Then, with $\underline{p}_u(m), \overline{p}_u(m)$ and $\underline{b}_u(m), \overline{b}_u(m)$ denoting the pressure and mass flow excess bounds for node u in operation mode m, the following inequalities are valid:

$$\sum_{m\in\mathscr{M}} \underline{p}_u(m)\, s_m \le p_u \le \sum_{m\in\mathscr{M}} \overline{p}_u(m)\, s_m \qquad \text{for all } u \in V, \tag{2}$$

$$\sum_{m\in\mathscr{M}} \underline{b}_u(m)\, s_m \le b_u \le \sum_{m\in\mathscr{M}} \overline{b}_u(m)\, s_m \qquad \text{for all } u \in V. \tag{3}$$

We call the model using the original operation modes the *compact model*, the one using fully specified operation modes the *extended model* and the extended model together with (2)–(3) the *bounded extended model*.

3 Topology Simplification for a Single Operation Mode

Our goal is to simplify the topology of a single operation mode of a compressor station to obtain a small "canonical" representation suitable for comparing operation modes via graph isomorphism detection (see Fig. 1).

We consider the network $N^m = (V, A^m, \underline{q}, \overline{q}, \underline{p}, \overline{p})$ corresponding to a fully specified operation mode m derived from the station network as follows. First, all closed elements are removed. Second, every shortcut, open valve and compressor in bypass is replaced by two opposing shortcuts with lower flow bound equal to zero. This is an equivalent transformation since the constraints for open valves or bypassed compressors are equivalent to those of shortcuts. Hence, the arc set A^m consists only of shortcuts and active compressors. Thus the model for a single operation mode becomes

$$0 \le \underline{p}_u \le p_u \le \overline{p}_u \qquad \text{for all } u \in V, \tag{4}$$

$$b_u = 0 \qquad \text{for all } u \in V_0, \tag{5}$$
$$\underline{q}_a = 0, \overline{q}_a = \infty \qquad \text{for all } a \in A_{\text{sc}}, \tag{6}$$
$$p_u = p_v \qquad \text{for all } (u,v) \in A_{\text{sc}}, \tag{7}$$
$$(p_u, p_v, q_a) \in P_a \subseteq \mathbb{R}^3_{\geq 0} \qquad \text{for all } (u,v) \in A_{\text{cg}}. \tag{8}$$

However, the network may be highly redundant, as a shortcut usually indicates that the incident nodes are identical. Thus we can reduce the size of the network by contracting a shortcut as follows. We identify the incident nodes of the shortcut and update the pressure bounds of the remaining node to be the intersection of the pressure intervals for the original nodes. If there are any other arcs between the two nodes, we do keep them as self-loops. But we need to be careful when applying this contraction since shortcuts sometimes do carry important information on the topology of feasible flows. We now devise a criterion for safely removing shortcuts. For this, we consider the shortcut subgraph of N^m, G^{sc}, its set of entries V^{sc}_+, its set of exits V^{sc}_- and for all entries $w \in V^{sc}_+$ the set $\vec{R}_N(w) \subseteq V^{sc}_-$ of exits reachable using only shortcuts:

$$G^{sc} := (V, A_{\text{sc}}) \tag{9}$$
$$V^{sc}_+ := V_\pm \cup \{w \in V \mid \exists u \in V : (u,w) \in A_{\text{cg}}\} \tag{10}$$
$$V^{sc}_- := V_\pm \cup \{w \in V \mid \exists u \in V : (w,u) \in A_{\text{cg}}\} \tag{11}$$
$$\vec{R}_{N^m}(w) := \{u \in V^{sc}_- : \exists\, w - u - \text{path in } G^{sc}\} \qquad \text{for all } w \in V^{sc}_+ \tag{12}$$

Proposition 1 *Consider a shortcut $\tilde{a} = (u,v)$ with $u \in V \setminus \{V^{sc}_+\}$ and the network N' arising from N when contracting $\tilde{a}$ to v. If*

$$\vec{R}_N(w) = \vec{R}_{N'}(w) \quad \textit{for all } w \in V^{sc}_+ \tag{13}$$

then for every admissible flow-pressure combination (p', q') for N' there exists an admissible flow-pressure combination (p, q) for N such that

$$q'_a = q_a \quad \textit{for all } a \in A_{\text{cg}}, \tag{14}$$
$$b'_w = b_w \quad \textit{for all } w \in V_\pm, \tag{15}$$
$$p'_w = p_w \quad \textit{for all } w \in V_\pm, \tag{16}$$

and vice-versa.

4 Computational Results

To investigate the effect of our method, we consider the compressor station network with three boundary nodes and four compressors shown in Fig. 1. We model the operating range $P_a = \{(p_u, p_v, q_a)\} \subseteq \mathbb{R}^3_{\geq 0}$ of each compressor $a = (u,v) \in A_{\text{cg}}$ by a

Table 1 Computational results on sample compressor station network. The first number is for feasible, the second number for infeasible instances

	Compact model	Extended model	Bounded extended model
Number of binary variables after presolve	29.6/33.5	34.1/33.2	27.6/33.2
Number of solving nodes	9.2/16.3	10.4 / 24.4	11.1/20.5
Presolving detected infeasibility	–/80.4%	–/80.1%	–/84.2%

simplified polyhedral model since we are only interested in the combinatorics of the compressor station model. In the original data there are 53 operation modes; these are used in the compact model. Enumerating all combinations of *active* and *bypass* for compressors leads to 655 fully specified operation modes. Removing infeasible operation modes and eliminating redundant modes using graph isomorphism detection after applying topology simplifications presented in Sect. 3 leaves 109 operation modes. These are used in our extended and bounded extended models.

We generated a large set of 58463 instances with varying flow amounts from one boundary node to one or both of the others at multiple different pressure levels, and checked whether each instance is feasible. We have used SCIP to solve our problems and the results showed that ca. 55% of the instances were feasible. To compare the performance of our extended models to the original compact one we consider the mean number of binary variables that have not been fixed by SCIP presolving and the mean number of branch-and-bound nodes required for solving. The solving times were negligible in all cases due to the absence of nonlinear constraints (see Table 1).

The results show that our preprocessing methods only have limited impact on the solver performance. We conjecture this to be due to the fact that we are considering the compressor station in isolation where combinatorics are simple enough for SCIP to perform well without further support. The next step is thus to apply our methods to optimizing large-scale gas networks.

References

1. Carter, R.G.: Compressor station optimization: computational accuracy and speed. Technical Report PSIG 9605. Pipeline Simulation Interest Group (1996)
2. Koch, T., Hiller, B., Pfetsch, M., Schewe, L (eds.): Evaluating gas network capacities. MOS-SIAM Series on Optimization. SIAM (2015)
3. Martin, A., Möller, M., Moritz, S.: Mixed integer models for the stationary case of gas network optimization. Math. Program. **105**(2), 563–582 (2006)
4. Rose, D., Schmidt, M., Steinbach, M.C., Willert, B.M.: Computational optimization of gas compressor stations: MINLP models versus continuous reformulations. Math. Methods Oper. Res. **83**(3), 409–444 (2016)
5. Ríos-Mercado, R.S., Borraz-Sánchez, C.: Optimization problems in natural gas transportation systems: a state-of-the-art review. Appl. Energ. **147**, 536–555 (2015)

6. van der Hoeven, T.: Math in gas and the art of linearization. Ph.D. thesis, Rijksuniversiteit Groningen (2004)
7. Wu, S., Ríos-Mercado, R.Z., Boyd, E.A., Scott, L.R.: Model relaxations for the fuel cost minimization of steady-state gas pipeline networks. Math. Comput. Model. **31**(2), 197–220 (2000)

Maximum Covering Formulation for Open Locating Dominating Sets

Blair Sweigart and Rex Kincaid

Abstract As a result of specific constructs and features, many graphs do not admit OLD-sets. We extend the traditional OLD-set definition by relaxing the dominating property. Instead of requiring all nodes to be covered by an OLD-set, we seek what we call a maximum covering OLD-set. Every graph has a maximum covering OLD-set. Furthermore, maximum covering OLD-sets allow an exploration of the tradeoff between the number of sensors placed and the number of nodes covered.

1 Introduction

Open-locating-dominating sets (OLD-sets) are a fault tolerant version of identifying codes, where we are able to detect and locate an event by examining the specific subset of nodes with "sensors" that indicate an event occurrence within their coverage range, but where the node at which the event occurred does not report. OLD-sets were first introduced in [9], with the motivation of intrusion detection sensor networks in buildings under the assumption that the intruder would render inoperable any sensor at the intrusion site. If the other sensors were able to detect intrusions at nearby points, then the OLD-set dictates where the sensors should be placed such that one can always determine the location of the intrusion, based on the specific sensors that indicate an event in their neighborhood. An OLD-set, as studied thusfar in the literature, must meet two criteria: (1) a *dominating* constraint where every node in the graph has at least one neighbor in the OLD-set and (2) a *locating* constraint where no two nodes in the graph have the same set of neighbors in the OLD-set [9]. We retain the common assumption of a detection radius of one, which provides that the neighborhood consists of adjacent nodes.

B. Sweigart (✉) · R. Kincaid
College of William & Mary, Williamsburg, VA, USA
e-mail: dbsweigart@email.wm.edu

R. Kincaid
e-mail: rrkinc@wm.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_35

Formally: in a graph G, with node set $V(G)$, edge set $E(G)$, and open neighborhoods $N(v) = \{w : vw \in E(G)\}, \forall v \in V(G)$, a set $\mathscr{D} \subseteq V(G)$ is an open locating dominating set if:

$$\forall v \in V,\ N(v) \cap \mathscr{D} \neq 0 \tag{1}$$

$$\forall v_1, v_2 \in V : v_1 \neq v_2,\ N(v_1) \cap \mathscr{D} \neq N(v_2) \cap \mathscr{D} \tag{2}$$

OLD-sets are similar to identifying codes [6], but act on the open, rather than closed neighborhoods. A further connection between OLD-sets and identifying codes is found in "strongly identifying codes" [5]. These codes can identify an event location regardless if the source node correctly self reports an event or faults; they work simultaneously on the open and closed neighborhood constructs. Work on identifying codes and strongly identifying codes includes applications and codes in different structures, but has largely focused on optimality bounds on codes. Most work on OLD-sets has examined minimum density sets on infinite grids [7] and minimum cardinality sets on finite graphs of certain structures [10]. A. Lobstein maintains a bibliography of papers on identifying codes and locating-dominating sets, with 349 entries as of September 2016 [8].

2 Maximum Covering Formulation

Most applications of OLD-sets require methods to quickly identify the set on a given graph. Finding an identifying code of minimum cardinality was shown to be NP-Hard [2]. Integer Programs (IP) have shown promise in identifying OLD-sets of minimum cardinality on arbitrary graphs [11]. The previous formulation in [11] used an adjacency matrix **A**, to satisfy the dominating constraint, then used a preconstructed node-pair matrix **B**, to satisfy the locating constraint. There are two primary limitations of this construct: the large number of graphs that permit no feasible OLD-set and the preconstructed nature of the **B** matrix.

While all graphs have [closed] locating-dominating sets [2], many graphs do not have OLD-sets [11]. This is due to specific graph constructs and the requirements of OLD-sets. Having two or more nodes in a graph that share the same open neighborhood is a sufficient condition for that graph to have no feasible OLD-sets.

Lemma 1 *In a feasible OLD-set, for every pair of nodes* $v_1, v_2 \in V(G), v_1 \neq v_2, \exists$ *some* $v_3 \in V(G) : v_3 \in \mathscr{D}, v_3 \in N(v_1), v_3 \notin N(v_2)$.

Proof Since $\mathscr{D}$ is feasible,we know $N(v) \cap \mathscr{D} \neq 0,\ \forall\ v \in V(G)$. Suppose $\nexists\ v_3 \in \mathscr{D} : v_3 \in N(v_1), v_3 \notin N(v_2)$. Then, $N(v_1) \cap \mathscr{D} = N(v_2) \cap \mathscr{D}$. But this contradicts the locating constraint, (2), in the definition of an OLD-set. □

Theorem 1 *If a graph has two or more nodes with the same neighborhoods* ($N(v_1) = N(v_2), v_1 \neq v_2$), *it has no feasible OLD-sets.*

Proof Assume there is a feasible OLD-set, $\mathscr{D} \in G$ and $\exists\ v_1, v_2 \in V(G) : N(v_1) = N(v_2), v_1 \neq v_2$. Since $\mathscr{D}$ is feasible, Lemma 1 provides: $\exists\ v_3 \in \mathscr{D} : v_3 \in N(v_1), v_3 \notin N(v_2)$. But then $N(v_1) \neq N(v_2)$ which contradicts the initial assumption. □

The reason for the infeasibility is clear if one examines a hub-and-spoke pattern, as often arise in scale-free graphs [1]. Consider a hub, node π, of degree $p \geq 3$, with r leaves, $2 \leq r < p$. Since these leaves have equivalent open neighborhoods, $N(v) = \{\pi\}$, by Theorem 1 there are no feasible OLD-sets.

If we relax the dominating condition of the OLD-set, we can define a modified OLD-set on any graph. That is, if we construct the OLD-set so that any node that is "covered" satisfies both the dominating and locating constraints, but any node not covered bears no effect on the selection of the OLD-set, we will be able to construct a "maximum covering" OLD-set. This is similar to the problem that frequently arises in network location theory literature of a similar name [3]. Under this construct, we seek to identify a minimum number of facilities that will maximize the demand that can be met (subject to some tradeoff function). We may also be limited in the number of facilities we can establish to meet demand and seek to identify the maximum demand (or number of nodes) that may be covered by a fixed number of facilities. This is known as the "fixed-p" maximum set covering problem [4].

A maximum covering OLD-set is a fundamental change in the OLD concept: we no longer require that every node be dominated.[1] In addition to the OLD-set $\mathscr{D}$, we introduce the "covered" set $\mathscr{C}$. To identify a maximum covering OLD-set, the integer program must dynamically consider constraints only when a given node is covered. The preconstructed nature of the **B** matrix does not permit this sort of dynamic inclusion, but the new $\boldsymbol{\Omega}$ construct, explained below, does. For flexibility, we have included weighting parameters, γ and ζ in the objective function to facilitate future tradeoff explorations. The improved formulation is:

$$\min\ \gamma \sum_{j \in V} c_j x_j - \zeta \sum_{i \in V} b_i y_i \tag{3}$$

$$\text{s.t.} \sum_{j \in V} A_{i,j} x_j \geq y_i \qquad \forall\ i \in V \tag{4}$$

$$\sum_{k \in V} (A_{i,k} - A_{j,k})^2 x_k \geq \Omega_{i,j} y_i y_j \qquad \forall\ i,j \in V \tag{5}$$

$$\sum_{j \in V} x_j \leq P \qquad \text{optional} \tag{6}$$

To implement the dynamic functionality we introduce a new variable y_i where:

$$y_i = \begin{cases} 1 & \text{if node } i \in \mathscr{C} \\ 0 & \text{otherwise} \end{cases}$$

[1] This actually slightly changes the definition of "dominated" since not all nodes need to have a neighbor in the OLD-set. The general idea holds, since the nodes that are "covered" are subject to the criteria, so we retain the terminology for consistency.

We again use the **A** matrix to satisfy the dominating constraint, Eq. 4. When $y_i = 0$, the right hand side (RHS) of the constraint is zero and removes any constraint on the selection of facilities x_j stemming from that node i.

In the locating constraint, Eq. 5, we introduce a $n \times n$ binary matrix $\boldsymbol{\Omega}$:

$$\Omega_{i,j} = \begin{cases} 1 & \text{if } 0 < SPD(i,j) \leq 2 \\ 0 & \text{otherwise} \end{cases}$$

where $SPD(i,j)$ is the shortest path distance between nodes i and j. If the SPD between the nodes is 1 or 2, the nodes are adjacent or share a neighbor, and $\Omega_{i,j} = 1$. The locating constraint must be enforced under the following conditions:

1. Nodes i and j share at least one neighbor ($\Omega_{i,j} = 1$)
2. Node $i \in \mathscr{C}$ ($y_i = 1$)
3. Node $j \in \mathscr{C}$ ($y_j = 1$)

If these three are met, then by Lemma 1, at least one facility location must distinguish the intersections of the neighborhoods with $\mathscr{D}$: The RHS will equal 1 and force the left hand side (LHS) to take on a value ≥ 1. If any of the three are NOT met, then RHS $\rightarrow 0$ and will impose no constraint on facility placement. We note that this constraint is now non-linear, but still binary.

The objective, max $|\mathscr{C}|$, may be set to max $\sum_i y_i \equiv \min(-\sum_i y_i)$. The minimization formulation may be directly combined with the min $|\mathscr{D}|$ objective to introduce a bi-objective optimization. To allow greater flexibility, we have introduced two weighting parameters γ and ζ to control the tradeoff between the cardinality objectives, as well as cost c_j and coverage value b_i parameters. We could also add an optional constraint, Eq. 6, to govern the maximum number of facilities to be placed as is traditionally found in network location theory literature [3].

3 Results

We offer three cases below that explore maximum-covering OLD-sets on various graphs, including examination of the tradeoff between min $|\mathscr{D}|$ and max $|\mathscr{C}|$, and a graph with no feasible OLD-set under the traditional construct.

Figure 1 shows the contrast between the traditional and maximum set covering formulations on a graph commonly used in OLD-set exploration [7, 9]. The fixed-p covering set, with p = 2, demonstrates the set $\mathscr{C}$. The traditional formulation requires $|\mathscr{D}| = 3$ to cover the graph. By limiting the number of facilities to 2, not all 5 nodes can be covered. The program identifies a maximum of $|\mathscr{C}| = 3$ nodes that can be covered by the OLD-set with $|\mathscr{D}| = 2$. We can verify the correctness of the maximum covering OLD-set by examining $N(v_i) \cap \mathscr{D}$.

- $N(v_1) \cap \mathscr{D} = v_3$.
- $N(v_2) \cap \mathscr{D} = v_1, v_3$.

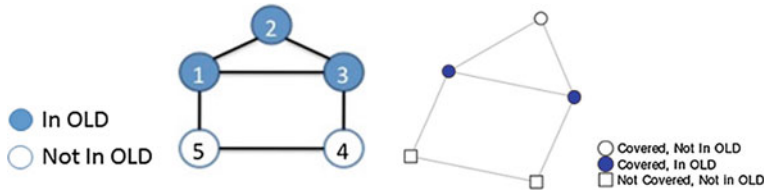


Fig. 1 The graph on the left depicts an OLD-set of the traditional formulation, while the graph on the right shows the an OLD-set with a fixed-p of 2

- $N(v_3) \cap \mathscr{D} = v_1$.
- $N(v_4) \cap \mathscr{D} = v_3$, which would be the same as v_1, so $v_4 \notin \mathscr{C}$.
- $N(v_5) \cap \mathscr{D} = v_1$, which would be the same as v_3, so $v_5 \notin \mathscr{C}$.

Figure 2 shows the tradeoff between min $|\mathscr{D}|$ and max $|\mathscr{C}|$. Specifically, all nodes of this graph can be covered with $|\mathscr{D}| = 8$, but various costs or weights may dictate a smaller number of nodes covered. We've shown here a fixed-p result with p = 6, $|\mathscr{C}| = 11$. There are several alternate optimal solutions. If we wished to impose that only nodes in $\mathscr{C}$ should be in $\mathscr{D}$, we could add a trivial constraint such as $y_i \geq x_i$.

As previously discussed, scale free graphs by their nature rarely have feasible OLD-sets. Figure 3 shows a maximum covering OLD-set on a 100 node, randomly

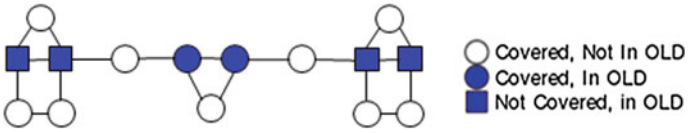


Fig. 2 Maximum Covering OLD-set on a 15 node graph with fixed p = 6

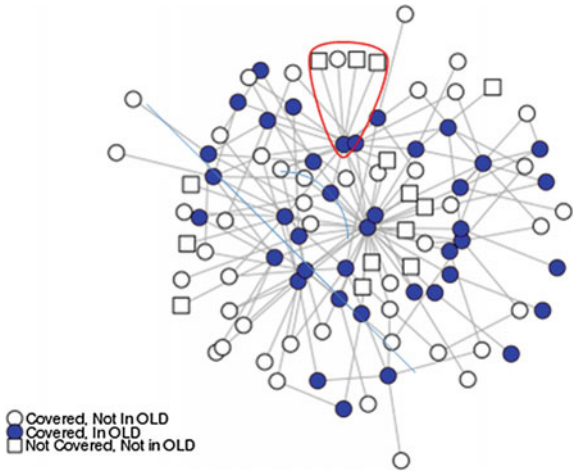


Fig. 3 Maximum Covering OLD-set on a 100 node, scale-free random graph. The area circled in red is a hub and spoke pattern

generated, scale-free graph. The OLD-set was identified using the formulation in Eqs. 3–5 with AMPL and a Gurobi 6.5 solver. Weights for Eq. 3 were set to maximize inclusion in $\mathscr{C}$: $\zeta = 10$, all others were set to 1. Note that near the top center of the graph is a hub and spoke pattern, circled in red. By Theorem 1 there is no way to define $\mathscr{D}$ to provide a unique intersection of each of the leaves' neighborhoods with $\mathscr{D}$ under the traditional OLD construct. The maximal covering OLD-set selects one of these four nodes for inclusion in $\mathscr{C}$ and excludes the others.

4 Further Research

Continued work will focus on two primary directions. The first is the exploration of computational efficiency. Specifically, we'll examine the the formulation presented here (with a linear objective function and a quadratic binary constraint) against formulations with various linearizations of the dominating constraint. We look to build upon the large body of work beginning with Fred Glover in the 1970s and continuing to present, though much of this research focuses on quadratic objective functions that are unconstrained or have linear constraints as in the quadratic assignment problem. The second focus will remain applications for OLD-sets. Areas that have shown promise include disease carrier identification, dark actor identification in adversary networks, and source financier identification in political networks.

References

1. Barabsi, A., Albert, R., Jeong, H.: Scale-free characteristics of random networks: the topology of the world-wide web. Phys. A Stat. Mech. Appl. **281**(1–4), 69–77 (2000)
2. Charon, I., Hudry, O., Lobstein, A.: Minimizing the size of an identifying or locating-dominating code in a graph is np-hard. Theoret. Comput. Sci. **290**(3), 2109–2120 (2003)
3. Church, R., Meadows, M.: Location modeling utilizing maximum service distance criteria. Geogr. Anal. **11** (1979)
4. Daskin, M.: Network and Discrete Location: Models, Algorithms, and Applications. Wiley (2011)
5. Honkala, I., Laihonen, T., Ranto, S.: On strongly identifying codes. Discrete Math. **254**(1), 191–205 (2002)
6. Karpovsky, M., Chakrabarty, K., Levitin, L.: On a new class of codes for identifying vertices in graphs. IEEE Trans. Inf. Theory **44**(2), 599–611 (1998)
7. Kincaid, R., Oldham, A., Yu, G.: Optimal open-locating-dominating sets in infinite triangular grids. Discrete Appl. Math. **193**, 139–144 (2015)
8. Lobstein, A.: Watching systems, identifying, locating-dominating and discriminating codes in graphs. http://perso.telecom-paristech.fr/~lobstein/debutBIBidetlocdom.pdf (2016)
9. Slater, P., Seo, S.: Open neighborhood locating-dominating sets. Aust. J. Comb. **46** (2010)
10. Seo, S., Slater, P.: Open neighborhood locating-dominating in trees. Discrete Appl. Math. **159**(6), 484–489 (2011)
11. Sweigart, D., Presnell, J., Kincaid, R.: An integer program for open locating dominating sets and its results on the hexagon-triangle infinite grid and other graphs. In: 2014 Systems and Information Engineering Design Symposium (SIEDS) (2014)

Part X
Health Care Management

Optimal Allocation of Operating Hours in Surgical Departments

Lisa Koppka, Matthias Schacht, Lara Wiesche, Khairun Bapumia and Brigitte Werners

Abstract A large part of revenue in hospitals is generated in surgical departments. In order to use available resources efficiently, we propose an innovative tactical optimization model to optimally allocate operating hours for operating rooms. An extensive simulation study is applied to evaluate the tactical plan with respect to main stakeholders. Results indicate strongly positive effects on staff and patients.

1 Introduction

Surgical interventions ensure for a large part of revenue in every hospital [3]. Considering scarce resources, thorough planning is highly important, especially for operating rooms (ORs). That applies particularly in heart centers, since almost every patient needs surgical intervention. Most approaches in OR planning deal with scheduling strategies and assume given capacities [2, 5]. Other approaches aim at allocating the same total capacities differently to influence main performance criteria [6]. It is possible to use provided resources differently without further expenditure, resulting in the same total capacity but different resource allocation. In close cooperation with a hospital for thoracic and cardiovascular surgery, we aim at determining optimal allocation of operation hours among ORs on a tactical level. Total operating time over all ORs is defined by available resources such as staff, equipment and legal regulations. We propose an innovative optimization model to optimally allocate operating hours. This tactical solution is evaluated using an extensive simulation study. Optimal allocation of total operating time with regard to different ORs and patients' requirements is able to positively affect staff and patients.

L. Koppka (✉) · M. Schacht · L. Wiesche · B. Werners
Faculty of Management and Economics,
Chair of Operations Research and Accounting,
Ruhr University Bochum, Bochum, Germany
e-mail: lisa.koppka@rub.de

K. Bapumia
Ruhr University Bochum, Bochum, Germany

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_36

2 Optimal Allocation of Operating Hours

In this chapter, we propose an optimization model for optimal allocation of operating hours. In hospitals, patients have different medical requirements and the ORs are differently equipped. Therefore, not every patient type can be treated in every OR. Furthermore, patient types are differentiated into patient groups according to their expected surgery duration. Depending on the specific hospital, the share of patients of each group varies considerably resulting in a hospital-specific case mix which is almost constant in the medium term. Since the demand for types of ORs varies dependent on the case mix, operating hours of the ORs need to match it. The following linear stochastic optimization model decides on daily operating hours for every OR on a tactical level which are valid for every day in the planning horizon.

On an operational level, the throughput of patients is an important performance criterion. For our medium term consideration, we take the number of patients to treat per day as given, to focus on the impact of operating hours. Another important criterion for hospital performance is employee satisfaction which needs to be high to guarantee best possible patient care. Hence, the objective is to minimize overtime as an indicator for staff satisfaction. Similar to common approaches to consider uncertainty in constraints, our model computes worst-case overtime. Taking the 95th percentile, the tactical model guarantees for a high probability of minimum overtime. Our tactical linear stochastic optimization model decides mainly on operating hours c_j in minutes for each specific OR $j \in \mathcal{J}$. Besides, patients are assigned to ORs depending on their medical group $l \in \mathcal{L}$ with the variable x_{jl}. Since we decide on a tactical level, all patients admitted must be assigned to an OR. Variation and uncertainty in the number of patients and type are represented through scenarios and reflect the hospital's case mix. Every scenario is weighted depending on the occurrence in the hospital's case mix with the parameter w^s. The objective is to minimize the expected sum of overtime minutes (o_j^{s+}) over all ORs $j \in \mathcal{J}$ and scenarios $s \in \mathcal{S}$ (see (1)).

$$\min \sum_{s \in \mathcal{S}} \sum_{j \in \mathcal{J}} w^s \cdot o_j^{s+} \tag{1}$$

The model considers that the overall OR time C per day remains constant and that in each scenario each patient is assigned to an OR (see (2–3)). c_j decides on the capacity of OR $j \in \mathcal{J}$ in minutes, while b_l^s is the number of patients of group $l \in \mathcal{L}$ to be scheduled in scenario $s \in \mathcal{S}$. x_{jl}^s decides on the number of patients of group $l \in \mathcal{L}$ to be treated in room $j \in \mathcal{J}$ in scenario $s \in \mathcal{S}$.

$$\sum_{j \in \mathcal{J}} c_j = C \tag{2}$$

$$\sum_{j \in \mathcal{J}} x_{jl}^s = b_l^s \qquad \forall l \in \mathcal{L}, s \in \mathcal{S} \tag{3}$$

Only one optimal allocation of operating hours for all scenarios is allowed and over- and undertime (o_j^{s+} or o_j^{s-}, respectively) are calculated through a worst-case assumption—every group's surgery duration is expected to be as long as the 95th percentile $a_l^{0.95}$ derived from historical data (see (4)).

$$\sum_{l \in \mathscr{L}} x_{jl}^s \cdot a_l^{0.95} + o_j^{s-} - o_j^{s+} = c_j \qquad \forall j \in \mathscr{J}, s \in \mathscr{S} \tag{4}$$

Moreover, it ensures that the assignments meet the requirements for every patient type (see (5) and Fig. 1). $\mathscr{A}$ is the set of patient groups with special requirements, and Eq. (5) prevents them from being assigned to an unsuitable OR.

$$\sum_{l \in \mathscr{L}_a} \sum_{j \in \mathscr{J} \setminus \mathscr{J}_a} x_{jl}^s = 0 \qquad \forall s \in \mathscr{S}, a \in \mathscr{A} \tag{5}$$

Finally, domain constraints ensure that $c_j \in \mathbb{N}_0$, $x_{jl}^s \in \mathbb{N}_0$ and $o_j^{s+}, o_j^{s-} \geq 0$. c_j is limited to an interval $\mathscr{C}$ defining minimal and maximal operating hours for the ORs. Since the operating hours are determined based on the considered scenarios, it is of high importance that these match the hospital's case mix. For computational reasons, it is not possible to include all information of daily patient occurrence provided by a hospital into the optimization model, instead, we use scenario reduction as in Heitsch and Römisch [4]. The Euclidean distance weighted inversely proportional with the average number of patients per week per group acts as a measure for the distance between two scenarios. The weighting parameter w^s is the share of scenarios best represented by scenario $s \in \mathscr{S}$. Implementation for heart centers is exemplarily shown in the next section.

3 Case Study

Data Analysis

This case study is based on data collected in a large hospital for thoracic and cardiovascular surgery in Germany. We include detailed data of more than 40,000 surgeries performed between 2009 and 2015. The patient collective consists of children patients, hybrid patients, who need combined cardiological and cardiothoracic interventions, and the remaining patients with no special requirements for the equipment of their OR (regular patients). These three patient types are additionally subdivided into nine patient groups according to their expected surgery duration (without emergencies). Figure 1 shows feasible assignments from patient groups to eight available ORs. There are three different types of ORs matching the patient types. Rooms one to six have no special equipment, room seven is a hybrid OR and room eight fits children's requirements. The total available OR time per day, $C = 5{,}340$ min, is allocated among the ORs and each room starts at 7:45 a.m. For every patient group (including

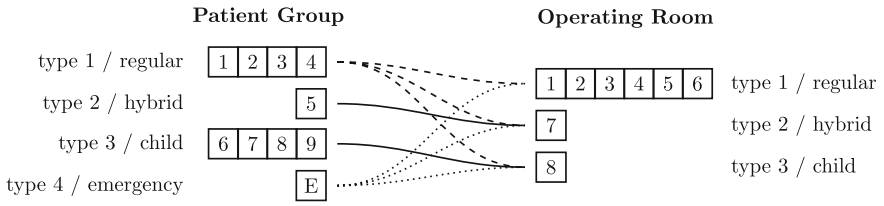


Fig. 1 Feasible assignments of patient groups to ORs

emergency patients), distribution functions for the surgery duration and the number of patients per week are calculated, which fit realistic data (see Table 1).

Strategies for Resource Allocation

The optimization model introduced in Sect. 2 computes optimal operating hours for the ORs. Using scenario reduction, 1,705 scenarios are reduced to ten best representing all remaining and thus representing the hospital's case mix. With this drastically reduced number of scenarios, information from 1,705 scenarios is aggregated in the weights and shapes of the remaining ten scenarios.

Figure 2 shows different alternatives for operating hours. Alternative ***OPT*** is the result of the optimization model, we compare it with the current real-world situation (Alternative ***R***) and with uniform operating hours (Alternative ***U***).

Evaluation

These three different alternatives are tested in a simulation study using our evaluation tool which simulates the workflow in the ORs (see Fig. 3). For each alternative, the corresponding operating hours are fixed for the whole investigation period. We measure the quality of the optimal results threefold with regard to three main stakeholders. Apart from the staff, we consider patients and management. On patient side, we study postponement of patients (actual treatment day $\neq$ planned treatment day). For management requirements we consider the OR utilization and on employee and especially physician side, we focus on the amount of overtime. Operational OR planning is often divided into two steps repeated daily or weekly. The first step plans the admission of patients for the upcoming week. Afterwards, the exact order of the surgeries is determined every day. Our evaluation tool is a framework that connects these planning steps by using the output of one step as the input for the following one. Each planning step is supported by hospital manager's rules. For example, children patients are preferably treated in the morning and patients whose appointments have been postponed from the day before are considered with higher priority in the next step to avoid further postponement. These daily schedules are included in our simulation model. The number of patients per week as well as the surgery duration for the patients is generated through probability distributions fitting realistic data and are considered according to Table 1. Moreover, the interarrival times of emergencies in minutes are exponentially distributed ($\lambda = 420$) to generate 24 emergencies per week on average. We evaluate 52 weeks, that is 52 runs of weekly admission planning

Table 1 Input data for the optimization and simulation models with 95th percentile (β), mean (μ) and deviation (σ)

			Regular				Hybrid	Child				Emergency
			1	2	3	4	5	6	7	8	9	E
Optimization	Surgery duration *(Empirical)*	β	171	291	356	446	256	121	260	385	529	395
Simulation	Surgery duration *(Shifted log-logistic)*	μ	76	186	241	294	154	72	139	237	312	188
		σ	62	79	58	73	42	22	89	80	107	206
	Number of patients *(Neg bin/bin)*	μ	25	20	32	9	1	1	2	2	4	15
		σ	8	6	9	3	2	1	1	1	2	7

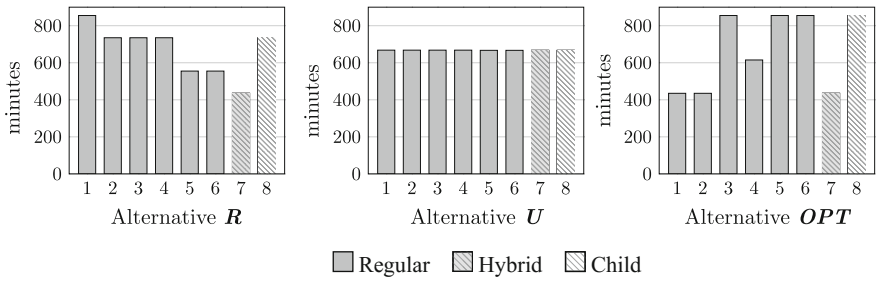


Fig. 2 Alternatives for allocation of operating hours

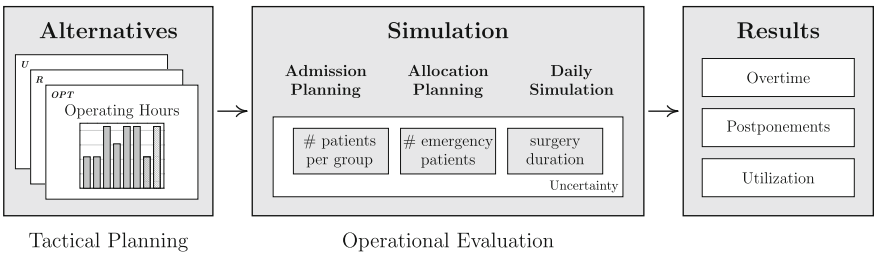


Fig. 3 Operational evaluation framework (Alternatives as in Fig. 2)

and 260 runs of daily allocation planning. During the year, we have approximately 5,400 patients undergoing surgery.

Results

Modified operating hours influence three main characteristics corresponding to the three main stakeholders in hospitals. Referring to staff interests, overtime should be avoided. Evaluation shows significant improvement in the daily amount of overtime over the whole period investigated. The 95th percentile of daily overtime—which is the sum of overtime minutes across all ORs—is 716.3 min for ***R***, 785.8 min for ***U*** and only 557.6 min for ***OPT***. As shown in the boxplots (Fig. 4), best results are achieved in ***OPT***, while ***U*** performs considerably worse. ***R*** shows similar results to ***OPT***, but each value, especially the median, is higher. Not only the staff, but also the patients benefit from an optimal resource allocation. ***OPT*** performs considerably better than ***U*** and ***R***. In ***OPT*** we have the highest share of patients being treated on the assigned day (94%). Compared to ***R*** (91% treated on the assigned day) the share of patients being postponed decreased by three percentage points (∼150 patients/year) using alternative ***OPT***. With 11% of postponed patients ***U*** performs worst. Although in general the stakeholder's interests are conflicting, our optimal solution supports the management's interests as well. As seen in Fig. 5, improving the working condition for the staff and patients' interests does not negatively affect the OR utilization or the number of patients being treated.

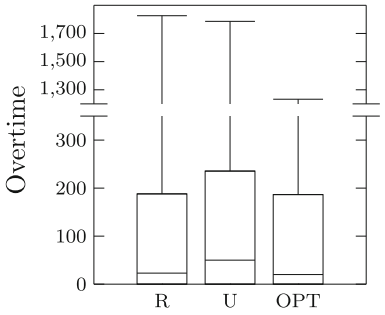


Fig. 4 Boxplots showing the daily amount of overtime in minutes

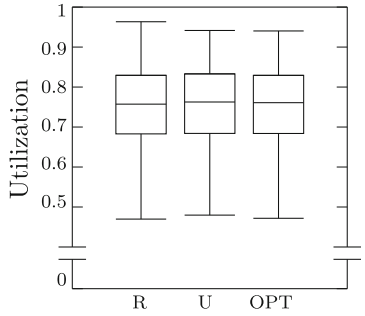


Fig. 5 Boxplots showing the daily utilization of the ORs

4 Conclusion

Using the innovative optimization model, staff overtime and patients rescheduling is considerably reduced. Reallocation of operating hours in ORs can promote main stakeholder's interests in a hospital. Variation of operating hours impacts shift planning and scheduling. Following the current development to flexible shift models in order to reconcile family and career, new operating hour schedules can be integrated to avoid unplanned overtime [1]. Further research on the optimal allocation of operating hours to investigate the interaction between over- and underestimation of surgery duration could additionally improve the performance of OR utilization.

References

1. Brunner, J., Bard, J., Kolisch, R.: Flexible shift scheduling of physicians. Health Care Manage. Sci. **12**(3), 285–305 (2009)
2. Denton, B., Viapiano, J., Vogl, A.: Optimization of surgery sequencing and scheduling decisions under uncertainty. Health Care Manage. Sci. **10**(1), 13–24 (2007)
3. Guerriero, F., Guido, R.: Operational research in the management of the operating theatre: a survey. Health Care Manage. Sci. **14**(1), 89–114 (2011)

4. Heitsch, H., Römisch, W.: Scenario tree modeling for multistage stochastic programs. Math. Programm. **118**(2), 371–406 (2009)
5. Hulshof, H.P.J., Kortbeek, N., Boucherie, J.R., Hans, W.E., Bakker, M.P.J.: Taxonomic classification of planning decisions in health care: a structured review of the state of the art in or/ms. Health Syst. **1**(2), 129–175 (2012)
6. Wiesche, L., Schacht, M., Werners, B.: Strategies for interday appointment scheduling in primary care. Health Care Manage. Sci. 1–16 (2016)

Part XI
Logistics, Routing and Location Planning

A Periodic Traveling Politician Problem with Time-Dependent Rewards

Deniz Aksen and Masoud Shahmanzari

Abstract The Periodic Traveling Politician Problem (PTPP) deals with determining daily routes for a party leader who holds meetings in various cities during a campaign period of τ days. On a graph with static edge costs and time-dependent vertex profits, PTPP seeks a closed or open tour for each day. The objective is the maximization of the net benefit defined as the sum of rewards collected from meetings in the visited cities minus the traveling costs normalized into a compatible unit. The reward of a meeting in a city are linearly depreciated according to the meeting date and recency of the preceding meeting in the same city. We propose a MILP formulation in which we capture many real-world aspects of the PTPP.

1 Introduction

In this paper we study the periodic traveling politician problem (PTPP) which is the generalization of the prize collecting traveling salesman problem (PCTSP). This problem can be considered as a Selective Multi-Period Dynamic Prize-Collecting TSP that asks for a closed or open tour maximizing the net benefit. In comparison to the classical TSP and PCTSP, PTPP contains extra elements of complexity. PTPP can be described as follows. Consider a set of cities $\mathbf{I} := \{0, 1, \ldots, n\}$ including a fictitious city (indexed as 0), a set of cities $\mathbf{I}' = \{1, \ldots, n\}$ including a campaign centre (capital city indexed as 1) and a set of days $\mathbf{T} = \{1, \ldots, \tau\}$ before the elections. On each day $t \in \mathbf{T}$, any city $i \in \mathbf{I}'$ can be visited either to hold a meeting there or without a meeting. A prize of BR_i (Base Reward) is specified for a meeting in each city $i \in \mathbf{I}'$ where the amount of BR_i depends on the population of city $i \in \mathbf{I}'$ and the ratio of votes of the politician's party (PP) in the previous election. We assume that $BR_i \geq 0$ for all $i \in \mathbf{I}'$. In addition, the actual reward earned by holding a meeting in city $i \in \mathbf{I}'$ on day t depends on two other factors: (i) The number of remaining

D. Aksen (✉) · M. Shahmanzari
Koç University, 34450 Sarıyer, Istanbul, Turkey
e-mail: daksen@ku.edu.tr

M. Shahmanzari
e-mail: mshahmanzari14@ku.edu.tr

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_37

days until the end of campaign, i.e. until the election day, denoted by $(\kappa - t)$. (ii) The number of days passed since the previous meeting in the same city, denoted by s where $1 \leq s \leq t-1$. The traveling cost between each pair of cities is known and given by $c_{ij}, i,j \in \mathbf{I}$ where c_{ij} denotes the cost of driving (or flying where applicable) from city i to city j. The traveling time between each pair of cities is also known and given by $d_{ij}, i,j \in \mathbf{I}$. The maximum tour duration to be observed while planning the tour of each day is denoted by ϕ. This time limit imposes an implicit threshold on the number of cities that can be visited in any given day. There is an explicit limit α on the number of daily meetings. The problem consists of selecting a subset of cities and visiting them with a tour starting and ending at city $i \in \mathbf{I}'$. Each city $i \in \mathbf{I}'$ is associated with a meeting time which is denoted by $\sigma_i, i \in \mathbf{I}'$. A distinctive feature of the PTPP is that there are three possible types of daily tours.

Type-1: Single-city tour. The politician wakes up in city i on day t, holds a meeting and sleeps in the same city i. We assume that the politician goes from city i to a fictitious city denoted by 0 and returns from 0 back to i. The travel costs and times between city 0 and any city $i \in \mathbf{I}'$ are zero in both directions.

Type-2: Multi-city closed tour. The politician wakes up in city i on day t, leaves i and visits at least one more city scheduled for that day. At the end of the day, he returns to the same city i to sleep that night.

Type-3: Multi-city open tour. The politician wakes up in city i on day t, leaves i and goes to another city j. In between i and j he may visit one or more cities, or he may directly travel from i to j. However, he does not return to i. Instead, he stays in j overnight, and wakes up there in the morning of day $(t+1)$.

Each city can accommodate at most one meeting a day. There can be an upper bound (such as three or four) on the total number of meetings held in each city during the campaign period. Moreover, the mechanism of the reward function will most likely prevent multiple visits to the same city in consecutive days. The durations of meetings range from 2 to 3 hours depending on the population of the host city. The cost of traveling by bus is 1.50 TL/km (Turkish Lira per Kilometer). For those cities with an airport and a bus travel time of more than 270 min from one another, we carefully check which travel option (airplane or party bus) is more time efficient. While doing this, we reckon with the commuting times between city centers and respective airports as well as with the check-in delays. The travel cost and time matrices are finalized after this investigation of the road travel times and flight times in Google Maps and TurkishAirlines.com. Finally, the politician cannot be away from the capital city Ankara (the campaign base) for more than κ consecutive days.

The existing literature of the PCTSP was surveyed in Feillet et al. [4]. Additionally, different methods are proposed in the literature to solve TSP and its variants in Brub et al. [2] and Cook [3]. Multi period TSP variants have been also studied in the literature. These problems usually deal with finding daily tours for a traveling salesman who provides a wide range of items to customers in different cities. Another significant variant of the TSP is the traveling purchaser problem with budget constraints [5].

2 Mathematical Modelling

In this section a mathematical formulation of the PTPP is proposed based on a MILP formulation for the TSP. The following additional index set are introduced besides the sets introduced earlier in Sect. 1: $\mathbf{T}' = \mathbf{T}\backslash\{1\}$: the set of all days of the campaign period excluding the first day. The following decision variables are introduced:
X_{ijt}: Binary variable indicating if arc (i,j) is traversed in day t $(i,j \in \mathbf{I}, t \in \mathbf{T})$.
L_{it}: Binary variable indicating if city i is not entered, but only left in day t.
E_{it}: Binary variable indicating if city i is not left, but only entered in day t.
S_{it}: Binary variable indicating if the politician sleeps in city i by the end of day t.
Z_{it}: Binary variable indicating if the politician holds a meeting in city i in day t.
FM_{it} : Binary variable indicating if the first meeting in city i is held in day t.
R_{its}: Binary variable indicating if city i accommodates two consecutive meetings in day t and day $(t-s)$ with no other meeting in between.
U_{it}: A continuous nonnegative variable used in the Modified Miller-Tucker-Zemlin Subtour Elimination Constraints determining the order of visit for city i in day t.
Given the index sets and the decision variables, PTPP can be formulated as follows:

$$\max .NET\_BENEFIT = \sum_{i\in I}\sum_{t\in T} BR_i \times \frac{\tau - t + 1}{\tau} \times FM_{it} + \sum_{i\in I}\sum_{t\in T}\sum_{1\le s<t} BR_i \times \frac{\tau - t + 1}{\tau} \times \frac{s}{k\tau} \times R_{its} - \sum_{i\in I}\sum_{j\in I}\sum_{t\in T} c_{ijt}X_{ijt} \tag{1}$$

$$\sum_{j\in I} X_{ijt} - \sum_{j\in I} X_{jit} = L_{it} - E_{it}, \quad i \in I, t \in T \tag{2}$$

$$L_{it} + E_{it} \le 1, \quad i \in I, t \in T \tag{3}$$

$$\sum_{j\in I} X_{ijt} \le 1, \quad i \in I, t \in T \tag{4}$$

$$\sum_{j\in I} X_{jit} \le 1, \quad i \in I, t \in T \tag{5}$$

$$S_{i(t-1)} \le S_{it} + \sum_{j\in I} \frac{L_{jt} + E_{jt}}{2}, \quad i \in I, t \in T' \tag{6}$$

$$\sum_{j\in I} \frac{L_{jt} + E_{jt}}{2} + S_{i(t-1)} \ge S_{it}, \quad i \in I, t \in T' \tag{7}$$

$$S_{i(t-1)} \le L_{it} + S_{it}, \quad i \in I', t \in T' \tag{8}$$

$$S_{0t} = 0, \quad t \in T \tag{9}$$

$$X_{i0t} = X_{0it}, \quad i \in I', t \in T \tag{10}$$

$$E_{it} \leq S_{it}, \quad i \in I', t \in T \tag{11}$$

$$S_{it} \leq \sum_{j \in I} X_{ij(t+1)}, \quad i \in I', t \in T, t \leq \tau - 1 \tag{12}$$

$$\sum_{k=t}^{t+\kappa} S_{1k} \geq 1, \quad t \in T,\ t \leq \tau - \kappa \tag{13}$$

$$Z_{it} \leq \sum_{j \in I} X_{ijt} + E_{it}, \quad i \in I',\ t \in T \tag{14}$$

$$Z_{it} \leq \sum_{j \in I} X_{jit} + L_{it}, \quad i \in I', t \in T \tag{15}$$

$$\sum_{j \in I'} Z_{jt} \leq \alpha, \quad t \in T \tag{16}$$

$$(n+1)S_{j(t-1)} + (n+1)(1 - X_{ijt}) + U_{jt} \geq U_{it} + 1, \quad i,j \in I, t \in T' \tag{17}$$

$$U_{it} \leq \sum_{j \in I} \sum_{k \in I} X_{jkt} + 1, \quad i \in I, t \in T \tag{18}$$

$$U_{it} \geq S_{i(t-1)}, \quad i \in I, t \in T' \tag{19}$$

$$U_{it} \leq S_{it} + \sum_{j \in I} X_{ijt}, \quad i \in I, t \in T \tag{20}$$

$$FM_{i1} = Z_{i1}, \quad i \in I' \tag{21}$$

$$FM_{it} \leq Z_{it}, \quad i \in I', t \in T' \tag{22}$$

$$FM_{it} \leq 1 - Z_{iu}, \quad i \in I', t \in T', 1 \leq u \leq t-1 \tag{23}$$

$$R_{its} \leq Z_{it}, \quad i \in I', t \in T', 1 \leq s \leq t-1 \tag{24}$$

$$R_{its} \leq Z_{i(t-s)}, \quad i \in I', t \in T', 1 \leq s \leq t-1 \tag{25}$$

$$\sum_{k=t-s+1}^{t-1} Z_{ik} \leq (s-1)(1-R_{its}), \quad i \in I', 3 \leq t \leq \tau, 2 \leq s \leq t-1 \tag{26}$$

$$R_{its} = 0, \quad i \in I, t \in T, t \leq s \leq \tau \tag{27}$$

$$R_{its} \leq 1 - FM_{it}, \quad i \in I', t \in T', 1 \leq s \leq t-1 \tag{28}$$

$$R_{ius} \leq 1 - FM_{it}, \quad i \in I', t \in T', t+1 \leq u \leq \tau, 1 \leq u-s \leq t-1 \tag{29}$$

$$\sum_{i \in I'} Z_{it}\sigma_i + \sum_{i \in I} \sum_{j \in I} X_{ijt} d_{ij} \leq \phi, \quad t \in T' \tag{30}$$

$$X_{ijt}, L_{it}, E_{it}, S_{it}, Z_{it}, FM_{it}, R_{its} \in \{0,1\}, \quad U_{it} \geq 0 \tag{31}$$

The objective function (1) seeks to maximize the difference between collected rewards and the incurred routing costs. Constraints (2) and (3) are coupling constraints between L, E and X. Constraints (4) and (5) are incoming and outgoing degree constraints. Note that these degree constraints are imposed as inequality due to the selective nature of PTPP. The politician does not have to visit or to hold a meeting in every city. Constraints (6)–(8) force the politician to sleep in the waking city for every day if there is a closed tour on that day. Constraints (9) and (10) prevent the politician from sleeping in fictitious city and force him to exit if he enters there. Constraints (11) force the politician to sleep in the last city of a Type-3 Tour. Constraints (12) ensure that he leaves the sleeping city next day. Constraints (13) enforce visits to the capital city every κ days. Constraints (14)–(16) are coupling constraints between Z and X. Constraints (17)–(20) are Modified MTZ subtour elimination constraints. Constraints (21)–(29) are coupling constraints between FM, Z and R. Constraints (30) guarantee that the daily maximum tour duration is not violated. Constraints (31) ensure binary integrality and nonnegativity, respectively.

3 Preliminary Computational Results

Four factors are considered in the calculation of the reward collected from a meeting in each city: (i) Population, (ii) Ratio of votes received by the PP in the previous election, (iii) Number of the remaining days until the election, and (iv) Number of days passed since the last meeting. Two factors, namely population and ratio of the PP votes in the previous election, directly affect the BR_i while the two other factors make the BR_i time dependent. BR_i is calculated as follows.

$BR_i = Criticality\_Factor_{(i)} \times (Base\_Reward_{(i)} + \frac{Population_{(i)}}{Min.Population} \times Population\_Multiplier)$

Table 1 Comparison of different solutions of the same instance with 39 cities and 15 days

Model	MILP solution	LP solution	Best possible	Relative gap (%)	CPU time (s)
Full MILP	21,146.5	–	27,713.5	23	86,435
LR	–	65,585.7	–	–	38
PLR1	26,125.3	–	29,143.9	10	3,791
SFLR	25,345.3	–	29,596.6	14	43,216
PLR2	57,431.7	–	58,744.2	2	35,537

Depending on the vote ratio of PP in the previous election, electoral zones can be divided into different criticality categories. Note that our aim is to come up with a base reward function that produces rewards not only according to city population, but also according to the criticality (importance) of that city. We considered 39 cities of Turkey with the highest BR_i values and a campaign period of $\tau = 15$ days.
The best feasible solution (the lower bound on the true optimal solution) of this problem is reported as 21,146.5. We used the commercial solver Gurobi 6.5.0 inside the mathematical modeling suite GAMS 24.6.2 on a Dell T3500 workstation with Intel Xeon W3960 processor. The CPU time limit was applied as 24 hours. Considering the 23% relative gap, we examined whether we can find a tighter upper bound. To this end, we investigated four types of relaxations: (i) Linear Relaxation of binary decision variables (LR), (ii) Partial Linear Relaxation of the binary routing variables X (PLR1), (iii) Semi-Full LP Relaxation with SL, FM and Z forced to be binary and all other originally binary variables relaxed between 0 and 1 (SFLR), and (iv) Partial Linear Relaxation of the binary variables S (PLR2). The comprehensive nonrelaxed model (denoted as FULL MILP) has 22,782 binary variables after the reductions performed by Gurobi at the root node before the iterations commence. By relaxing the binary variables X, this number reduced to 2,885 for the Partial Linear Relaxation (PLR1) version. Table 1 presents the test results obtained from the five models. The final upper bound for PLR1 is 29,143.9 which is worse than the final upper bound of the MILP model, namely 27,713.5. The lower bound of SFLR at the end of 12 h is as high as 57,431.7 and the upper bound is 58,744.2. These are extremely loose bounds. Therefore, the FULL MILP upper bound (the best feasible solution) is probably the tightest bound we could obtain so far. In this solution, the number of meetings held is 43, and the number of cities visited is 39. The values of the total collected rewards and total travel costs are 33,824 and 15,663 respectively.

4 Concluding Remarks

In this paper we introduced a MILP formulation for the Periodic Traveling Politician Problem with time-dependent rewards. It can be viewed as a multi-period version of the prize-collecting traveling salesman problem with dynamic profits, arbitrary depot

nodes, and three types of time restricted tours. The objective function seeks to maximize the collected rewards minus the traveling costs by visiting a subset of cities at each day subject to maximum tour duration. Many real-life aspects are incorporated into the formulation of the problem. To examine the performance of the proposed MILP model, several relaxation schemes have been tested. PTPP is a new problem in the literature. Our study will stimulate other researchers to work on this rigorous problem which could open new gates in election logistics.

References

1. Applegate, D.L., Bixby, R.E., Chvatal, V., Cook, W.J.: The Traveling Salesman Problem: A Computational Study. Princeton University Press (2011)
2. Brub, J.F., Gendreau, M., Potvin, J.Y.: A branch and cut algorithm for the undirected prize collecting traveling salesman problem. Networks **54**(1), 56–67 (2009)
3. Cook, W.: In Pursuit of the Salesman: Mathematics at the Limits of Computation. Princeton University Press, Princeton, USA (2011)
4. Feillet, D., Dejax, P., Gendreau, M.: Traveling salesman problems with profits. Transp. Sci. **39**(2), 188–205 (2005)
5. Labadie, N., Mansini, R., Melechovský, J., Calvo, R.W.: The team orienteering problem with time windows: an LP based granular variable neighborhood search. Eur. J. Oper. Res. **220**(1), 15–27 (2012)

An Emission-Minimizing Vehicle Routing Problem with Heterogeneous Vehicles and Pathway Selection

Martin Behnke, Thomas Kirschstein and Christian Bierwirth

1 Introduction

In addition to cost and time, greenhouse gas (GHG) emissions have become a further command variable for planning processes in the transportation industry. A large number of scientific literature is devoted to the development of planning approaches taking into account the emissions of transport processes. In order to minimize a transport process' emissions, many factors affecting emissions have been studied. Besides the total distance covered by a transport process, the modal split, payload, traveling speed as well as vehicle type and specifications are identified as most influential planning parameters.

In this paper, we present an emission-oriented vehicle routing problem with heterogeneous vehicles and pathway selection (EVRP-VC-PS). The model seeks to find a set of tours such that the total emission quantity of all vehicles employed is minimized and all customers are served while the vehicles' load restrictions are met. We use an emission model taking into account vehicle-specific and road-specific characteristics as well as payload as parameters, see [3].

In an experimental study, we sketch a network structure typically found in city logistics where a set of customers in an urban area is to be served from a suburban depot via urban or highway pathways. We examine the effects of different

M. Behnke · T. Kirschstein (✉) · C. Bierwirth
School of Economics and Business, Martin-Luther-University,
Gr. Steinstraße 73, 06108 Halle, Germany
e-mail: thomas.kirschstein@wiwi.uni-halle.de

M. Behnke
e-mail: martin.behnke@wiwi.uni-halle.de

C. Bierwirth
e-mail: christian.bierwirth@wiwi.uni-halle.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_38

objective functions (namely emission, time, and distance minimization) and pathways w.r.t. total emissions, total travel time, and total travel distance. Furthermore, we study the effects of different network layouts by varying the highway radius.

2 An Emission-Oriented Vehicle Routing Problem with Heterogeneous Vehicles and Pathway Selection

In contrast to the classical VRP, we consider the possibility to choose between different pathways connecting two nodes, so we obtain a graph in which parallel arcs are possible, see e.g. [1]. We model these pathways as a set $A_{ij} := \delta^{+i} \cap \delta^{-j}$ of directed arcs from i to j, where δ^{+i} refers to the outgoing arcs and δ^{-i} to the ingoing arcs of i. For each arc we assume constant distance, acceleration and speed.

The estimation of GHG emissions during a vehicle tour is calculated with the mesoscopic emission model presented in [3]. The factors influencing GHG emissions are driving speed, acceleration frequency, load, distance, and technical vehicle specifications. The resulting emission function can be simplified into a linear function with a load-dependent and a load-independent emission factor depending on arc characteristics (such as distance, speed, etc.) and vehicle specifications.

For our research, we consider four different vehicle types with payload capacities of 2.5, 5.5, 14 and 25 tons and different vehicle parameters compiled from [2] and [4] (details can be found on http://prodlog.wiwi.uni-halle.de/forschung/research_data/).

Using the emission model designed by Kirschstein and Meisel [3] and the notation of Table 1, we obtain the following MILP for the EVRP-VC-PS

Table 1 Variables and parameters of the linear program

Variables			
x_{ak}	Number of vehicles $k \in K$ using arc a	l_{ak}	Load of vehicle $k \in K$ on arc $a \in A$
Parameters			
C	Set of customers: $C := \{1, \dots, n\}$	V	Set of all vertices: $V := \{0, \dots, n\}$
δ^{+i}	Set of outgoing arcs of $i \in V$	δ^{-i}	Set of ingoing arcs of $i \in V$
A_{ij}	Set of arcs from i to j: $A_{ij} := \delta^{+i} \cap \delta^{-j}$	A	Set of all arcs: $A := \bigcup_{i,j \in V} A_{ij}$
K	Set of vehicle types: $K := \{1, \dots, m\}$	d_a	Distance of arc $a \in A$
q_i	Demand of customer $i \in C$	v_k	Available vehicle number of type $k \in K$
$cap^{\max}$	Maximum load capacity of all vehicles	c_{ak}^{fix}	Load-independent emission coefficient
m_k^{tare}	Tare weight of vehicle type k	c_{ak}^{load}	Load-dependent emission coefficient

$$\min \rightarrow \sum_{a \in A} d_a \sum_{k \in K} c_{ak}^{\text{fix}} \cdot x_{ak} + c_{ak}^{\text{load}} \cdot \left(m_k^{\text{tare}} \cdot x_{ak} + l_{ak} \right) \tag{1}$$

s.t.

$$\sum_{a \in \delta^{-j}} \sum_{k \in K} x_{ak} = 1, \qquad \forall j \in C \tag{2}$$

$$\sum_{a \in \delta^{+i}} \sum_{k \in K} x_{ak} = 1, \qquad \forall i \in C \tag{3}$$

$$\sum_{a \in \delta^{-j}} x_{ak} = \sum_{a \in \delta^{+j}} x_{ak}, \qquad \forall j \in C, k \in K \tag{4}$$

$$\sum_{a \in \delta^{+0}} x_{ak} = v_k, \qquad \forall k \in K \tag{5}$$

$$\sum_{a \in \delta^{+j}} \sum_{k \in K} l_{ak} = \sum_{a \in \delta^{-j}} \sum_{k \in K} l_{ak} - q_j, \qquad \forall j \in C \tag{6}$$

$$l_{ak} \leq cap_k \cdot x_{ak}, \qquad \forall a \in A, k \in K \tag{7}$$

$$\sum_{a \in \delta^{+0}} \sum_{k \in K} l_{ak} = \sum_{i \in C} q_i \tag{8}$$

$$\sum_{a \in \delta^{-0}} \sum_{k \in K} l_{ak} = 0 \tag{9}$$

$$x_{ak} \in \mathbb{N}_0,\ l_{ak} \in \mathbb{R}^+ \qquad a \in A, k \in K \tag{10}$$

The objective is to minimize the total amount of emissions produced by all vehicle types. In the experiments, we also test the model with distance and time minimization objective, i.e.

$$\sum_{a \in A} d_a \sum_{k \in K} x_{ak} \qquad \text{and} \qquad \sum_{a \in A} \tau_a \sum_{k \in K} x_{ak}$$

where τ_a is the travel time of arc $a \in A$.

Restrictions (2) and (3) ensure to visit and leave each customer exactly once, (4) guarantee that the vehicle type does not change during the tour. For every vehicle type k, (5) ensure that exactly v_k vehicles leave the depot. Vehicles not required for deliveries immediately return to the depot on arc $(0, 0)$. With restriction (6) we ensure that the load leaving a customer equals the load reaching it reduced by its demand. The compliance of the maximum vehicle capacity is secured by (7). The following two restrictions constitute valid cuts. The sum of all customer demands has to be transported (8) and no load reaches the depot (9), (10) are the domain declarations.

3 Construction of Instances

3.1 Creation of Data Sets

To test the proposed model, we split Solomon's R1 instance into consecutive chunks of 15 customers obtaining six different instances. Each instance is interpreted as an urban area. Next to urban roads, we introduce a highway circle around the center of the customer nodes, i.e. the average of the customer coordinates $C = \left(x^C, y^C\right)$. The circle's radius r is varied between $\{0.5,\ 0.7,\ 1.0,\ 1.2\}$ of the distance from the center C to the farthest customer location. In any constellation, the depot is placed on the most northern point of the highway $(x^C, y^C + r)$.

We calculate two types of arcs for each pair of nodes $i, j \in V$, so every A_{ij} consists of exactly two elements. Type $t = U$ represents a shortest-distance path $a := (i, j, U) \in A_{ij}$ through the city using urban roads calculated as the Euclidean distance between the coordinates of customers i and j, i.e. $d_{ijU} := d_2(i, j)$. For urban roads, low average speed and high frequency of acceleration processes is assumed. Path type $t = H$ denotes the highway path $a := (i, j, H) \in A_{ij}$. Here, the vehicle drives the shortest way from i to the highway (point $P1$), then drives on the highway to the point $P2$ closest to node j. Afterwards, the vehicle travels the remaining way through the urban city area. That is $d_{ijH} = d_2(i, P1) + d_{circ}(P1, P2) + d_2(P2, j)$ with $d_{circ}(i, j) = \pi \cdot r \cdot \frac{\alpha(P1,P2)}{180^\circ}$ of the circular arc and the enclosed angle $\alpha(P1, P2)$ defined by C, $P1$, and $P2$. For the segment traveled on the highway $d_{circ}(P1, P2)$, a higher average speed and less frequent acceleration processes are assumed. Finally, we divide the demands of the Solomon instances by 4 in order to fit to the supposed vehicle capacities. In total, we obtain $6 \cdot 4 = 24$ test settings.

3.2 Preprocessing

To reduce the complexity of the test instances, we check for any pair $i, j \in V$ if one of the arcs (i, j, U) and (i, j, H) is dominating the other. The dominated arc is then deleted from the instance. Obviously, in case of distance and time minimization for any pair of nodes $i, j \in V$ only one arc is efficient.

If emission minimization is considered, the decision is more complex. An arc can only be deleted if and only if emissions are less or equal on one arc for all vehicles and all possible loads. In Table 2 it is shown that for 0.65–10.29% of all pairs, both arcs are kept after preprocessing.

Table 2 Average values under emission optimization dependent on the highway radius

Radius	Emissions	Emission	Runtime	Share of highway arcs		Kept double arcs (%)
	w/o highway	w/ highway	in s	in E (%)	in solution (%)	
0.5	43.13	39.85	19.85	40.38	34.02	10.29
0.7	46.85	42.17	34.12	35.04	40.57	7.03
1.0	55.14	51.88	77.29	7.70	17.64	1.43
1.2	61.31	60.75	140.82	1.42	6.68	0.65

4 Results of Computational Study

The constructed instances are solved with ILOG CPLEX 12.6.3.0 on a 64-bit Windows 10 Pro system (Intel Core i7-2600, 8 GB memory).

4.1 Effect of Pathway Selection

Table 2 displays the average characteristics of the optimal solutions for the instance sets categorized by highway radii. As expected, total emissions increase with increasing radius and, thus, increasing distance to the depot. Likewise, runtimes increase, too, despite the fact that less arcs remain after preprocessing. Hence, it looks like instances with a higher distance scale are easier to solve. Furthermore, the smaller the highway radius the less arcs are deleted by preprocessing. Although, the share of highway arcs used in the optimal solutions reaches a maximum of about 40% for $r = 0.7$. Overall, emissions can be reduced by about 1–10% when considering highway pathways. Again, relative savings are maximal at a radius of $r = 0.7$.

4.2 Effect of Objective

Here, we compare the solutions under emission, travel distance, and travel time optimization. For studying the joint performance under the different objectives, Fig. 1 shows the average total emissions, total traveled distances, and total travel times relative to the minimum of each measure. Figure 1 reveals that emission minimization produces the most unbalanced pattern with about 25 and 13% increase in travel distance and travel time, respectively. In contrast, the most leveled performance pattern is obtained by time minimization showing an increase of about 12% for distance and emissions as well. Distance optimization shows the highest surcharge in emissions and an equal increase in travel time as for emission optimization.

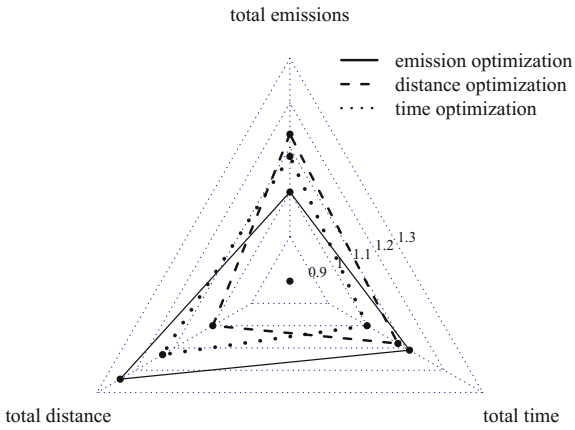


Fig. 1 Relative amount of performance indicators under different objective functions

Table 3 Average characteristics of optimal solutions under different objective functions

	Emission optimization	Distance optimization	Time optimization
Employed vehicles	4.74	3.38	3.44
# highway arcs	4.88	0	4.89
# urban arcs	14.86	18.38	13.55
Runtime (s)	62.26	7.79	7.04

Table 3 shows the average number of vehicles as well as the number of highway and urban arcs used in the optimal solutions. Additionally, the average runtimes are reported. It appears that emission minimization is more difficult to solve as the average runtime is almost 10 times higher than compared with distance and time optimization. Most probably the increased solving effort is caused by the fact that load weight contributes to the objective. Emission minimization also employs more vehicles and, as a consequence, uses more arcs in total. The share of used highway arcs is highest for time optimization, as expected.

To summarize this paper highlights that considering alternative routes in vehicle routing problem particularly for urban areas may allow decision makers to reduce GHG emissions. The proposed emission-orienting vehicle routing model was tested on artificial data sets of small scale with only two pathways. Nevertheless, relevant savings in GHG emissions could be generated by considering both pathway options. In practice, however, many more alternative pathways may exist such that (a) the emission reduction potential is much larger and (b) more sophisticated solution methods are required in order to solve large instances.

Acknowledgements This research is funded by the German Research Foundation (DFG) under reference BI 555/4-1.

References

1. Garaixa, T., Artigues, C., Feillet, D., Josselin, D.: Vehicle routing problems with alternative paths: an application to on-demand transportation. Eur. J. Oper. Res. **204**, 62–75 (2010)
2. Hausberger, S., Rexeis, M., Zallinger, M., Luz, R.: Emission Factors from the Model PHEM for the HBEFA Version 3, Graz (2009)
3. Kirschstein, T., Meisel, F.: GHG-emission models for assessing the eco-friendliness of road and rail freight transports. Transp. Res. Part B **73**, 13–33 (2015)
4. Scora, G., Barth, M.: Comprehensive Modal Emissions Model (CMEM), version 3.01, User's Guide. http://www.cert.ucr.edu/cmem/docs/CMEM_User_Guide_v3.01d.pdf (2006). Accessed 04 June 2016

Window Fill Rate in a Two-Echelon Exchangeable-Item Repair-System

Michael Dreyfuss and Yahel Giat

Abstract The fill rate service measure describes the proportion of customers who commence service immediately upon arrival. Since, however, customers will usually tolerate a certain wait time, managers should consider the window fill rate in lieu of the fill rate. That is, the performance measure of interest is the probability that a customer is served within the tolerable wait time. In this paper, we develop approximation formulas for the window fill rate in a two-echelon, exchangeable-item repair system in which the upper echelon is a central depot and the lower echelon comprises multiple locations. We demonstrate the use of the formulas through a numerical example and measure the approximation error of the window fill rate formulas using simulation.

1 Introduction

Exchangeable-item repair systems are systems to which customers bring a failed item and exchange it for a serviceable item. We consider a two-echelon system similar to [10] that comprises multiple locations in the lower echelon and a central depot in the upper echelon. The repair facilities in the lower echelon are capable of repairing only certain failures. If a failure cannot be repaired on-site, the item is shipped to the central depot for repair. To improve the system's performance, spares may be allocated to each of the locations.

In this paper, the system's performance is an extension of the fill rate measure. The fill rate assumes that customers penalize the firm *if* they wait. In most cases, however, customers will tolerate a certain period of wait and therefore the firm does not incur reputation costs if the customer waits less than the tolerable wait. In [7], the fill rate measure is extended to incorporate this customer patience in a single-echelon setting, and termed as the *window fill rate*, that is, the probability that the

M. Dreyfuss (✉) · Y. Giat
Jerusalem College of Technology, Jerusalem, Israel
e-mail: dreyfuss@jct.ac.il

Y. Giat
e-mail: yahel@jct.ac.il

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_39

customer is served *within* the tolerable wait. The goal of this paper is to extend these results and develop approximating formulas for the window fill rate in a two-echelon system.

Our paper contributes to the research of multi-echelon exchangeable-item repair systems originated by Sherbrook's METRIC model [15] that develops an approximate evaluation of the number of backorders in a multi-echelon system and describes a greedy algorithm to solve the spares allocation problem. This body of research is presented in books such as [14, 16] and recently reviewed in [2]. Except for the fact that we limit the system to a two-echelon structure, we assume the standard METRIC assumptions, which include ample repair servers, that components fail according to a Poisson process with a constant arrival rate and a continuous $(S-1, S)$ review policy.

Many METRIC-based papers focus on the number of back-orders performance measure (e.g., [1, 6, 9]) or the fill rate performance measure (e.g., [4, 13]). The disadvantage of the fill rate is that it does not take into account research such as [8] that report that customers will tolerate a certain period of wait, (see also [11] who use the term "reasonable duration"). We incorporate this, by considering the *window* fill rate, i.e., the probability of a random customer to be served within a certain time window. Berg and Posner [3] develop a mathematical expression for the window fill rate for a single location and [7] characterize its functional form and develop an algorithm to find the near optimal spares allocation in a multiple location single echelon mode. We extend these papers to a two-echelon system.

2 The Model

The system comprises two echelons with L locations in the lower echelon and one central depot in the upper echelon. We number the depot as location $l = 0$ and the lower-echelon locations as $l = 1, .., L$. Customers arrive to each of the $L + 1$ locations at rate $\lambda_l \geq 0,\ l = 0, ..., L$. Each location has ample identical repair servers with i.i.d. repair times. The local repair facilities are able to repair only certain failures whereas more complex failures are sent for repair in the central depot. In each location, the cumulative repair time distribution is given by $G_l(\cdot)$. Customers receive the available items according to a first-come first-serve policy. To reduce customer waiting time, the system keeps a number of spares, so that if there is a spare item available in stock it is given immediately to the client in exchange for the arriving failed one. After receiving an item, the customer leaves the system. Let p_l denote the probability that an item can be repaired on-site and is not forwarded to the depot. Lateral shipments are not allowed and therefore the probability of being forwarded to the depot from location $l > 0$ is $1 - p_l$. The central depot itself is able to repair all types of failures and therefore $p_0 = 1$. The back and forth shipment times from any lower-echelon location l to the central depot are i.i.d. with a probability density function $d_l(t)$.

The replenishment time is the time between a customer's arrival to a location until the customer's item or its replacement joins the location's stock. In the lower

echelon, replenishment could happen in one of two ways. With probability p_l, repair is done on-site and therefore the probability for replenishment within time t is $G_l(t)$. Alternatively, with probability $1 - p_l$, an order is opened and forwarded with the failed item to the central depot and returns with a serviceable item. In this case, for replenishment to happen within time t, the waiting time at the depot plus shipment time must be less than t. Therefore, in the lower echelon

$$\begin{aligned} &Pr[\text{replenish} \le t] = \\ &\quad = Pr[\text{repair here}]Pr[\text{repair} \le t] + Pr[\text{repair at depot}]Pr[\text{shipment + wait at depot} \le t] \\ &\quad = p_l G_l(t) + (1 - p_l) \int_{x=0}^{t} Pr[\text{shipment} = x]Pr[\text{wait at depot} \le t - x]dx \\ &\quad = p_l G_l(t) + (1 - p_l) \int_{x=0}^{t} d_l(x) F_0(s_0, t - x)dx. \end{aligned}$$

In the above, $F_0(s_0, t)$ is the window fill rate of the depot, that is, the probability that a customer or order arriving to the depot is served within t units of time when there are s_0 spares in the depot. In contrast to the lower echelon locations, in the depot, replenishment happens only due to repair on-site. Therefore, the cumulative distribution of the replenishment time, $R_l(s_0, t)$, is given by:

$$R_l(s_0, t) = \begin{cases} p_l G_l(t) + (1-p_l) \int_{x=0}^{t} d_l(x) F_0(s_0, t-x)dx, & \text{if } l = 1, ..., L \\ G_0(t), & \text{if } l = 0. \end{cases} \tag{1}$$

The total arrival rate at the central depot, $\hat{\lambda}_0$, is the sum of the customers arriving to it and the orders that are forwarded to it from the lower echelon. Thus, $\hat{\lambda}_0 = \lambda_0 + \sum_{l=1}^{L}(1 - p_l)\lambda_l$. For the other locations, $l = 1, ..., L$, the total arrival rate is equal to the customer arrival rate $\hat{\lambda}_l = \lambda_l$.

Since arrivals to the depot are independent, if the depot has no spares then the replenishment times are also independent (recall, the depot has ample servers and each server repair time is i.i.d.). In contrast, when there are spares in the depot the replenishment times described above are dependent (see [10]). In the ensuing analysis, we follow the standard METRIC model [15] approach and neglect this dependency. In Sect. 3 we measure the error due to neglecting the dependency by comparing the window fill rates derived by the approximation formulas with the window fill rates computed through simulation.

Proposition 1 *If location l is allotted s spares then the window fill rate for tolerable time t, $F_l(s, t)$ is given by*

$$F_l(s, t) = P[\hat{Y}_l(t) \le s - 1] + R_l(s_0, t)P[\hat{Y}_l(t) = s]. \tag{2}$$

where $\hat{Y}_l = Y_{1,l} - Y_{2,l}$ *and where* $Y_{1,l}$ *and* $Y_{2,l}$ *are Poisson random variables with parameters* $\hat{\lambda}_l \int_{u=t}^{\infty}(1 - R_l(s_0, u))du$ *and* $\hat{\lambda}_l \int_{u=0}^{t} R_l(s_0, u)du$*, respectively.*

Proof Follows immediately from Proposition 9 of [3], where we replace the repair time with the replenishment time and the arrival rate with the total arrival rate. □

Plugging (1) into (2), $F_l(s,t)$ is given by

$$F_l(s,t) = \begin{cases} P[\hat{Y}_l(t) \le s-1] \\ \quad + \left(p_l G_l(t) + (1-p_l)\int_{x=0}^{t} d_l(x) F_0(s_0, t-x)dx\right) P[\hat{Y}_l(t) = s] & \text{if } l = 1, \ldots, L \\ P[\hat{Y}_l(t) \le s-1] \\ \quad + G_0(t) P[\hat{Y}_l(t) = s] & \text{if } l = 0. \end{cases} \tag{3}$$

Proposition 2 *a.* $F_l(s,t)$ *is increasing with s.*
b. $F_l(s,t)$ *either concave or convex-concave with s.*

Proof Follows from the proof of Proposition 1 of [7]. The difference here is that we have the replenishment time, $R_l(s_0, t)$, instead of the repair time, $G_l(t)$. However, since the proof does not make any assumptions on the functional form of the repair distribution function, it is permissable to replace $R_l(s_0, t)$ with $G_l(t)$. □

Let $\mathbf{s} = (s_0, ..., s_L)$ denote the spares allocation in the system (depot and lower-echelon locations). The *system's* window fill rate, $F(\mathbf{s}, t)$, is the weighted average of window fill rates in each node as follows,

$$F(\mathbf{s}, t) = \sum_{l=0}^{L} \frac{\lambda_l}{\lambda} F_l(s_l, t). \tag{4}$$

where $\lambda = \sum_{l=0}^{L} \lambda_l$ is the sum of customer arrivals to the system.

3 Numerical Example

We demonstrate the uses of the window fill rate formulas (3)–(4) using parametric values loosely based on a small-scale realistic problem. Consider a depot serving four locations ($L = 4$) in which all customers tolerate a wait of $t = 9$. For simplicity, we assume that all the lower-echelon locations are identical in their arrival rate and probability for on-site repair. Specifically, $\lambda_l = 0.06, p_l = 0.5$ for all $l = 1, \ldots, 4$. We assume, however, that customers do not arrive directly to the depot, that is, arrivals to the depot are only by repairs forwarded from the lower-echelon locations. Thus, $\lambda_0 = 0$ and p_0=1. Repair time in all the locations (including the depot) is normally distributed with mean 45 and standard deviation 10. Travel time from each location to the depot is constant, $d_l = 5$.

For this example, we assume that there are $S = 12$ spares available for operations. In Table 1 we report the window fill rate for different spares allocations. We consider s_0 values of 0, 4, 8, 12. When there are twelve spares in the depot, there is only one possible lower-echelon allocation. For eight, four and zero spares in the depot there are five, fifteen and thirty-three *different* lower-echelon allocations, respectively (recall, the lower-echelon locations are identical). For each s_0 that we consider, we report at most four allocations; the allocation with the highest and lowest window fill rates.

To test the accuracy of the window fill rate formulas, we compare the formula-derived window fill rate with the window fill rate values that are derived through simulation. For each simulation, one thousand independent replications were simulated each for time durations equivalent to 10,368 demand events. The average of the one thousand observed window fill rates is reported. In all our simulations, the half width of the 95% confidence interval is less than 0.00081.

Recall, we neglect the replenishment dependency. As discussed in Sect. 2, when there are no spares in the depot the window fill rate formulae are accurate. Indeed, by Table 1, when $s_0 = 0$ the absolute error is less than 0.1%, which is within the confidence interval. In contrast, when there are spares in the depot, the dependency of the replenishes increases and the formulas are less accurate (see upper rows of Table 1). Furthermore, the number of replenishes decreases with p_l. Indeed, the error is most appreciable when $p_l = 0$, i.e., when all the items are repaired at the depot (see the right column of Table 1). Of the fifty-four allocations that we examine, when $p_l = 0.5$ the absolute error never exceeds one percent. When $p_l = 0$ the errors of

Table 1 The formula-based and simulation-based window fill rate for different spares allocation

		$p_l = 0.5$		Error[a] (%)	$p_l = 0$		Error[a] (%)
Allocation		Window fill rate			Window fill rate		
s_0	$(s_1, , s_4)$	Formulas (%)	Simulation (%)		Formulas (%)	Simulation (%)	
12	(0, 0, 0, 0)	21.50	21.06	0.4	67.15	71.54	4.4
8	(4, 0, 0, 0)	38.36	38.21	0.1	36.75	42.36	5.6
8	(3, 1, 0, 0)	45.50	45.42	0.1	48.55	51.27	2.7
8	(2, 1, 1, 0)	50.39	50.41	0.0	58.84	58.57	0.3
8	(1, 1, 1, 1)	51.81	51.92	0.1	64.43	62.62	1.8
4	(8, 0, 0, 0)	28.16	29.11	1.0	25.24	25.90	0.7
4	(7, 1, 0, 0)	34.54	35.45	0.9	31.04	32.20	1.2
4	(3, 2, 2, 1)	59.75	59.59	0.2	55.05	55.61	0.6
4	(2, 2, 2, 2)	62.30	61.96	0.3	57.33	57.74	0.4
0	(12, 0, 0, 0)	25.00	25.01	0.0	25.00	25.02	0.0
0	(11, 1, 0, 0)	27.48	27.50	0.0	27.13	27.17	0.0
0	(3, 3, 3, 3)	59.34	59.40	0.1	55.41	55.47	0.1
0	(4, 4, 4, 0)	59.80	59.86	0.1	57.46	57.50	0.0

[a]The absolute difference between the formula-derived and the simulation-derived window fill rates

only nine allocations exceed one percent and the maximal absolute error is 5.6%. The advantage of using formulas over the more accurate simulation is in the computation time. Whereas the time to execute each simulation is approximately thirty minutes, the time to compute the formulas-derived window fill rate is instantaneous.

4 Conclusions

In this paper, we derive approximating formulas for the window fill rate in a two-echelon exchangeable-item repair-system. We compare the formula-derived window fill rate values with the window fill rate that is computed through simulation. For the specific example that we use we find that the maximal absolute error is 5.6%. This example demonstrates that in many situations, the cost in terms of loss of accuracy may be negligible compared to the gains in computing time. For example, if managers are seeking to maximize the window fill rate then many evaluations are needed and using the formulas is preferable to simulation the window fill rate.

References

1. Basten, R.J.I., van Houtum, G.J.: Near-optimal heuristics to set base stock levels in a two-echelon distribution network. Int. J. Prod. Econ. **143**(2), 546–552 (2013)
2. Basten, R.J.I., van Houtum, G.J.: System-oriented inventory models for spare parts. Surv. Oper. Res. Manage. Sci. **19**(1), 34–55 (2014)
3. Berg, M., Posner, M.J.M.: Customer delay in $M/G/\infty$ repair systems with spares. Oper. Res. **38**(2), 344–348 (1990)
4. Caggiano, K.E., Jackson, P.L., Muckstadt, J.A., Rappold, J.A.: Optimizing service parts inventory in a multiechelon, multi-item supply chain with time-based customer service-level agreements. Oper. Res. **55**(2), 303–318 (2007)
5. Diaz, A., Fu, M.C.: Models for multi-echelon repairable item inventory systems with limited repair capacity. Eur. J. Oper. Res. **97**(3), 480–492 (1997)
6. Dreyfuss, M., Giat, Y.: Multi-echelon exchangeable-item repair system optimization. Working Paper. Jerusalem College of Technology (2016)
7. Dreyfuss, M., Giat, Y.: Optimal spares allocation in an exchangeable-item repair system with tolerable wait. Eur. J. Oper. Res. forthcoming (2017)
8. Durrande-Moreau, A.: Waiting for service: ten years of empirical research. Int. J. Serv. Ind. Manage. **10**(2), 171–194 (1999)
9. Ghaddar, B., Sakr, N., Asiedu, Y.: Spare parts stocking analysis using genetic programming. Eur. J. Oper. Res. (forthcoming) (2016)
10. Graves, S.C.: A multi-echelon inventory model for a repairable item with one-for-one replenishment. Manage. Sci. **31**(10), 1247–1256 (1985)
11. Katz, K.L., Larson, B.M., Larson, R.C.: Prescription for the waiting-in-line blues: entertain, enlighten, and engage. MIT Sloan Manage. Rev. **32**(2), 44 (1991)
12. Levner, E., Perlman, Y., Cheng, T.C.E., Levner, I.: A network approach to modeling the multi-echelon spare-part inventory system with backorders and interval-valued demand. Int. J. Prod. Econ. **132**(1), 43–51 (2011)
13. Lien, R.W., Iravani, S.M., Smilowitz, K.R.: Sequential resource allocation for nonprofit operations. Oper. Res. **62**(2), 301–317 (2014)

14. Muckstadt, J.A.: Analysis and Algorithms for Service Parts Supply Chains. Springer Science & Business Media (2005)
15. Sherbrooke, C.C.: METRIC: a multi-echelon technique for recoverable item control. Oper. Res. **16**(1), 122–141 (1968)
16. Sherbrooke, C.C.: Optimal Inventory Modeling of Systems: Multi-Echelon Techniques, vol. 72. Springer Science & Business Media (2004)

Redistricting in Mexico

Miguel Ángel Gutiérrez-Andrade, Eric Alfredo Rincón-García, Sergio Gerardo de-los-Cobos-Silva, Antonin Ponsich, Roman Anselmo Mora-Gutiérrez and Pedro Lara-Velázquez

Abstract Redistricting is the redrawing of the boundaries of legislative districts for electoral purposes in such a way that Federal or state requirements are fulfilled. In 2015 the National Electoral Institute of Mexico carried out the redistricting process of 15 states using a nonlinear programming model where population equality and compactness were considered as conflicting objective functions, whereas other criteria, such as contiguity, were included as constraints. In order to find high quality redistricting plans in acceptable amounts of time, two automated redistricting algorithms were designed: a Simulated Annealing based algorithm, and an Artificial Bee Colony inspired algorithm. Computational results prove that the population based technique is more robust than its counterpart for this kind of problems.

1 Introduction

The zone design problem arises from the need of aggregating small geographical units (GUs) into regions, in such a way that one (or more) objective function(s) is (are) optimized and some constraints are satisfied. The design of electoral zones or electoral redistricting is a well known case, due to its influence in the results of

M.Á. Gutiérrez-Andrade · S.G. de-los-Cobos-Silva · P. Lara-Velázquez
Universidad Autónoma Metropolitana Iztapalapa, Mexico City, Mexico
e-mail: gamma@xanum.uam.mx

S.G. de-los-Cobos-Silva
e-mail: cobos@xanum.uam.mx

P. Lara-Velázquez
e-mail: plara@xanum.uam.mx

E.A. Rincón-García (✉) · A. Ponsich · R.A. Mora-Gutiérrez
Universidad Autónoma Metropolitana Azcapotzalco, Mexico City, Mexico
e-mail: rigaeral@correo.azc.uam.mx

A. Ponsich
e-mail: aspo@correo.azc.uam.mx

R.A. Mora-Gutiérrez
e-mail: mgra@correo.azc.uam.mx

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_40

electoral processes and its computational complexity, which has been shown to be NP-Hard [1]. In this framework, the GUs are grouped into a predetermined number of zones or districts, and democracy must be guaranteed through the satisfaction of restrictions that are imposed by law. In particular, some generally proposed criteria are population equality, to ensure the "one man one vote" principle; compactness, to avoid any unfair manipulation of the border or shape of electoral zones for political purposes, and contiguity, to prevent from designing fragmented districts [2, 5, 6].

In Mexico, in 2015, the National Electoral Institute (INE) started the design of new redistricting plans for 15 federal entities using two automated redistricting algorithms: a Simulated Annealing (SA) based algorithm and an Artificial Bee Colony (ABC) inspired algorithm. The primary purpose of this paper is to describe the main characteristics of these algorithms. To address this issue, we provide a description of the problem in Sect. 2. A brief overview of the inner working mode of the SA, and ABC algorithms are presented in Sects. 3 and 4 respectively. Some computational results are detailed in Sect. 5. Finally, some conclusions and perspectives for future work are drawn in Sect. 6.

2 Problem Description

In Mexico, there are two different types of districts used to elect federal and local representatives. In 2015, a process to produce new local redistricting plans for 15 federal entities started, and two heuristic based algorithms were used. These algorithms seek for a redistricting plan that represents the best balance between population equality and compactness.

In order to promote population equality of a district Z_s, the following measure was used:

$$C_1(Z_s) = \left(\frac{1 - \left(\frac{P_{Z_s}}{P_M} \right)}{0.15} \right)^2 \tag{1}$$

where P_{Z_s} is the population of district Z_s, and P_M is the state average population. Each district Z_s is defined through a set of binary variables x_{is} such that $x_{is} = 1$ if the ith GU belongs to district Z_s and $x_{is} = 0$ otherwise. Finally, 0.15 is the maximum percentage of deviation allowed, 15%.

To promote compactness, a metric that can be easily computed, and requires low computation time was used. This measure compares the perimeter of a district Z_s with that of a square having the same area.

$$C_2(Z_s) = \left(\left(\frac{PC_{Z_s}}{\sqrt{AC_{Z_s}}} * 0.25 \right) - 1 \right) \tag{2}$$

where PC_{Z_s} and AC_{Z_s} are the perimeter and the area of the considered district Z_s, respectively. Thus, districts with a good compactness will have a compactness value close to 0.

In order to handle the multi-objective nature of the problem, a weight aggregation function strategy was used:

$$Minimize\, f(P) = \Sigma_{i=1}^{n} \lambda_1 C_1(Z_i) + \lambda_2 C_2(Z_i) \tag{3}$$

where P is a redistricting plan, $P = \{Z_1, Z_2, \ldots, Z_n\}$, with a predefined number of districts, n. The weighting factors were established after a discussion between political parties and INE's authorities. Both sectors agreed that the main objective in this process is to preserve the principle "one man one vote", even above the shape of the districts. Thus, population equality was considered twice as important as compactness, and the weighting factors were set to $\lambda_1 = 1$ and $\lambda_2 = 0.5$. Finally, the construction of redistricting plans is subjected to constraints that guarantee that (**R1**) each district is connected, (**R2**) a predefined number of districts, n, is constructed, and (**R3**) each GU is assigned to exactly one district.

Since the design of electoral zones is an NP-Hard problem, the automated heuristic algorithms are an appropriate strategy to design electoral redistricting plans. In the following sections, we give a brief description of the SA and ABC algorithms used in the redistricting process in Mexico.

3 Simulated Annealing Adaptation

Simulated Annealing is a metaheuristic introduced by Kirkpatrick in [4]. We implemented a classical version of SA, with a geometric decreasing cooling schedule.

The initial solution is created using the following strategy. All GUs are labelled as available. The algorithm then selects randomly n GUs, assigns them to different districts and labels them as not available. Finally, each district is iteratively extended by adding an available GU having a frontier with the district in its current shape. Each GU incorporated to a district is labelled as not available in order to avoid the construction of overlapping districts. The latter step is performed until all the GUs are labelled as not available. This way, the initial solution satisfies constraints **R1**–**R3**. Note that SA and ABC use the same procedure to create initial solutions.

For the construction of a neighbour solution SA uses the following strategy. A random district, Z_i, is chosen and a GU in this district is moved to a neighbour district, Z_j. If this move produces a disconnection in district Z_i, the following repair process is applied. The number of connected components in Z_i is counted, and the connected component that has the bigger number of GUs is defined as district Z_i; subsequently, the remaining components are assigned to Z_j.

The new solution is evaluated and accepted or rejected according to the Metropolis criterion. This process is repeated until the temperature reaches a predefined lower bound.

4 Artificial Bee Colony Adaptation

Artificial Bee Colony (ABC) is a bio-inspired metaheuristic, originally proposed by Karaboga [3]. However, the ABC heuristic was originally designed for continuous optimization problems. In order to handle discrete decision variables, we implemented some modifications to the ABC algorithm based on a recombination strategy.

First, M food sources are generated using the strategy described in Sect. 3. The number of onlooker and employed bees is set equal to the number of food sources, and exactly one employed bee is assigned to each food source. Then, each employed bee, i, modifies its food source, P_i, applying the strategy used by SA described in Sect. 3. If the new solution, V_i, has a nectar amount better than or equal to that of P_i, V_i replaces P_i and becomes a new food source exploited by the hive. In other case, V_i is rejected and P_i is preserved.

As soon as the employed bees process has been completed, each onlooker bee chooses two solutions. The first solution, P_1, is randomly selected depending on a probability associated with the objective function cost. The second solution, P_2, is randomly selected from the food sources exploited by the hive. A new food source, V_1, is produced through a recombination technique described straightforward.

A GU k is randomly selected. Thus, there is a district $Z_i \in P_1$ and a district $Z_j \in P_2$ such that $k \in Z_i \cap Z_j$. Let us now consider the following sets: $H_1 = \{l : x_{li} = 0, x_{lj} = 1\}$ and $H_2 = \{l : x_{li} = 1, x_{lj} = 0\}$. Then a GU in H_1 is inserted into Z_i, and a GU in H_2 is extracted from Z_i, and inserted into any randomly chosen district contiguous to Z_i.

If these moves produce a disconnection in district Z_i, the following repair process is applied. The algorithm defines the connected component of Z_i that includes GU k (i.e., the GU used within the above-described recombination strategy) as district Z_i; subsequently, the remaining components are assigned to other adjacent districts. In this way, properties **R1**–**R3** are preserved.

The new solution, V_1, is accepted or rejected using the greedy selection process applied by employed bees.

5 Experimental Results and Discussion

The two algorithms described in the previous section were already applied in the local redistricting process of 15 states. The remaining 17 states will be redistricted during 2016. 100 runs of both algorithms were executed in each state, and the bests solutions were proposed as the new redistricting plans.

In Table 1 we present for each algorithm the best cost found, mean cost, standard deviation, average running time per run in minutes, and the average evaluations of the objective function (EOF) for each state. In addition, we performed a Wilcoxon test to prove if solutions produced by both algorithms are significantly different. We tested the null hypothesis that the medians of the costs of both algorithms are identical. If the null hypothesis was accepted a value 0 was assigned to both algorithms, which

Table 1 Costs for both algorithms

State	Algorithm	Best cost	Mean cost	Std. dev.	Avg time	EOF	WT
Aguascalientes	ABC	5.4186	5.7425	0.0918	5.7727	7,451,003.33	0
	SA	5.3302	5.7246	0.2193	3.1244	8,640,835.83	0
Baja California	ABC	3.8336	4.0459	0.0874	18.2508	7,709,056.29	1
	SA	3.8883	4.3452	0.1773	8.8736	7,376,802.67	−1
Chihuahua	ABC	8.553304	8.7280	0.07320	17.1777	5,773,122.75	−1
	SA	8.462909	8.8679	0.2047	7.6237	7,491,716.25	1
Coahuila	ABC	7.2210	7.2868	0.0261	8.9151	7,905,368.31	1
	SA	7.5175	7.9816	0.2013	5.2421	10,885,648.74	−1
Durango	ABC	6.8811	6.9527	0.0309	8.1486	7,598,931.17	1
	SA	7.0959	7.4913	0.2015	5.2493	9,913,248.13	−1
Hidalgo	ABC	11.5753	11.5753	0.00	2.5871	3,718,736.35	1
	SA	11.5753	11.7334	0.1462	1.3799	4,154,362.50	−1
Nayarit	ABC	9.7177	9.8053	0.0370	5.9575	7,520,319.49	1
	SA	9.7200	9.9513	0.1290	4.4475	9,615,257.22	−1
Oaxaca	ABC	12.8909	13.5953	0.2748	6.2958	7,630,291.47	−1
	SA	12.4440	13.5566	0.6023	3.2547	8,952,958.51	1
Puebla	ABC	13.1271	13.6825	0.1923	9.5846	7,747,284.92	1
	SA	13.1411	14.0174	0.3741	5.2023	11,125,132.00	−1
Quintana Roo	ABC	5.3823	5.5770	0.0724	8.4673	7,632,413.53	−1
	SA	5.2671	5.5276	0.1249	4.9709	9,044,919.33	1
Sinaloa	ABC	10.1996	10.4231	0.1057	25.7972	7,650,348.36	1
	SA	10.3392	11.1804	0.4071	12.4396	5,868,655.20	−1
Tamaulipas	ABC	7.4608	7.5200	0.0301	16.5675	7,577,753.14	1
	SA	7.8316	8.0775	0.1148	10.0377	7,721,859.47	−1
Tlaxcala	ABC	9.8962	9.8962	0.00	2.2129	7,408,774.25	0
	SA	9.8962	9.9059	0.0314	1.7790	8,946,774.25	0
Veracruz	ABC	21.8584	62.4072	70.8789	10.3452	7,583,854.67	0
	SA	20.2893	49.3686	24.7216	4.7909	7,968,780.56	0
Zacatecas	ABC	9.1067	9.1089	0.0021	7.6998	7,457,025.83	1
	SA	9.1469	9.2427	0.0288	4.5083	6,452,574.58	−1

represents that both strategies exhibit the same behaviour. If the null hypothesis was rejected a value of −1, or 1, was assigned to the algorithm with the lower or higher median respectively. These results are presented in column 8 of Table 1.

Results in Table 1 highlight that for most of the states the ABC algorithm has lower standard deviation than the SA version. On the other hand, the average runtime per run used by the ABC algorithm was always higher than the time used by SA. However, INE considered that this difference can be omitted since the redistricting plans produced will be used for 3 years. Finally, using the Wilcoxon test we can conclude that ABC outperforms the SA version, since ABC was able to generate, on average, lower cost solutions in 9 of the 15 states, while SA only excelled in 3 states.

6 Conclusions

In this paper, we presented a Simulated Annealing based algorithm, and an Artificial Bee Colony based algorithm used by INE, to realize the local electoral redistricting process of 15 states in 2015. Both heuristic algorithms solve the optimization problem corresponding to the redistricting process, promoting the design of compact districts with the same amount of inhabitants. In order to compare the performance of the proposed algorithms we applied a Wilcoxon test and concluded that both techniques had a similar performance in 3 states. However, ABC was able to outperform SA in 9 of the remaining states. Thus, we can say that on average ABC will have a better performance than SA in redistricting problems with similar criteria to those described in this paper. On the other hand, SA always was faster than ABC.

In 2017 the federal redistricting process will begin, and INE wishes to propose an algorithm that improves the results obtained so far. Therefore, further research include the creation of an optimization algorithm that improves the performance of ABC and SA.

References

1. Altman, M.: Is automation the answer: the computational complexity of automated redistricting. Rutgers Comput. Law Technol. J. **23**, 81–141 (1997)
2. Bozkaya, B., Erkut, E., Laporte, G.: A tabu search heuristic and adaptive memory procedure for political districting. Eur. J. Oper. Res. **144**, 12–26 (2003)
3. Karaboga, D., Basturk, B.: A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. J. Glob. Optim. **39**, 459–471 (2007)
4. Kirkpatrick, S., Gellat, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220**, 671–680 (1983)
5. Ricca, F., Scozzari, A., Simeone, B.: Political districting: from classical models to recent approaches. J. Oper. Res. **204**(1), 271–299 (2011)
6. Rincón-García, E.A., Gutiérrez-Andrade, M.A., de-los-Cobos-Silva, S.G., Lara Velázquez, P., Mora-Gutiérrez, R.A., Ponsich, A.: ABC, a viable algorithm for the political districting problem. In: Gil-Aluja, J., Terceño-Gómez, A., Ferrer-Comalat, C.J., aMerigó-Lindahl, M.J., Linares-Mustarós, S. (eds.) Scientific Methods for the Treatment of Uncertainty in Social Sciences, vol. 377, pp. 269–278. Springer International Publishing (2015)

Min-Max Fair Emergency System with Randomly Occupied Centers

Jaroslav Janáček and Marek Kvet

Abstract This paper deals with the min-max fair emergency service system design under uncertain ability of service providing. Studied generalized system disutility follows the idea that the individual user's disutility comes from more than one located center. We present here an advanced approximate algorithm for the min-max optimal emergency service system design, which is based on a radial formulation and valid exposing structures. Presented method enables its simple implementation within common optimization environment instead of special software development.

1 Introduction

This paper deals with emergency service system design, which belongs to the discrete network location problems [6]. The min-max fair public service system design problem proved to be easily solvable [1] by an iterative bisection method, when users' perceived disutility is proportional to the distance between a user and the nearest service center. The success of the above approach was based on the so-called radial formulation of the p-median problem [2] and the p-dispersion problem [7]. When limited ability of a service center providing users by service is considered, such situation may occur that a current demand of a user cannot be satisfied from the nearest service center due to the center is occupied by a demand, which has risen recently. The latter demand is then serviced from the second nearest center, etc. Thus user's disutility depends on distances from r nearest centers and it can be described as a weighted sum of the distances. Contrary to the average user's disutility objective, the first attempts at min-max fair emergency system design by minimizing an upper bound of all perceived generalized disutility values have failed [5]. The turn to better was achieved by introducing and applying so-called lexicographically minimal

J. Janáček (✉) · M. Kvet
Faculty of Management Science and Informatics, University of Žilina,
Univerzitná 8215/1, 010 26 Žilina, Slovak Republic
e-mail: jaroslav.janacek@fri.uniza.sk

M. Kvet
e-mail: marek.kvet@fri.uniza.sk

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_41

exposing structures [4]. Even if the lexicographically minimal exposing structure in combination with the generalized radial formulation of the design problem yielded very good solution in acceptable computational time, the solution was not strictly optimal. Within this paper, we enhance the searching process for lexicographically minimal exposing structure by an extension to obtain the optimal exposing structure, which assures the optimal design of the min-max fair emergency system.

2 Radial Formulation for the Generalized Disutility and Exposing Structure

To formulate a mathematical model of the min-max optimal emergency service system design problem, we denote the set of users' locations by J and the set of possible service center locations by I. The strategic decisions in the problem concern deployment of p centers in the set I of possible locations. The contribution to disutility of a user located at $j \in J$ provided by a center located at $i \in I$ is denoted by d_{ij}. The generalized disutility for any user is modeled by a sum of weighted disutility contributions from the r nearest centers. The weights q_k for $k = 1, \ldots, r$ are positive real values which meet the inequalities $q_1 \geq q_2 \geq \cdots \geq q_r$. We assume that the disutility contribution value ranges only over non-negative integers from $[0, m]$. The values divide the range into m zones. Let us define $v = m - 1$ for brevity of further expressions. To describe the system of radii formed by the values [3], a system of zero-one constants is defined so that the constant a_{ij}^s is equal to 1 if the disutility contribution d_{ij} for a user j from the possible center location i is less than or equal to s, otherwise a_{ij}^s is equal to 0. The location variables $y_i \in \{0, 1\}$ for $i \in I$ model the decision of service center location at i by the value of 1. In addition, we introduce auxiliary zero-one variables x_{jsk} for $j \in J, s \in [0 \ldots v], k \in [1 \ldots r]$ to model the disutility contribution value of the k-th nearest service center to the user j. The variable x_{jsk} takes the value of 1 if the k-th smallest disutility contribution for the customer $j \in J$ is greater than s and it takes the value of 0 otherwise. The associated model follows.

$$Minimize \qquad h \tag{1}$$

$$Subject\ \ to: \qquad \sum_{i \in I} y_i \leq p \tag{2}$$

$$\sum_{k=1}^{r} x_{jsk} + \sum_{i \in I} a_{ij}^s y_i \geq r \qquad for\ \ j \in J,\ \ s = 0, \ldots, v \tag{3}$$

$$\sum_{k=1}^{r} q_k \sum_{s=0}^{v} x_{jsk} \leq h \qquad for\ \ j \in J \tag{4}$$

$$x_{jsk} \in \{0,1\} \ \textit{for} \ j \in J, \ \ s = 0,\ldots,v, \ \ k = 1,\ldots,r; \ \ y_i \in \{0,1\} \ \textit{for} \ \ i \in I; \ \ h \geq 0 \tag{5}$$

The constraint (2) limits the number of located centers by p. The constraints (3) ensure that the sum of variables x_{jsk} over $k \in [1 \ldots r]$ expresses the number of the service centers outside the radius s from the user location j, which remains to the number r. The link-up constraints (4) ensure that each perceived disutility is less than or equal to the upper bound h. It was found that branch and bound method performs very slowly due to link-up constraints (4). The turn to better was achieved by applying the exposing structures. An exposing structure can be described by the triple $[u, S, G]$, which satisfies the following rules. The first component u is a positive integer less than or equal to r. The second component S is an u-tuple $[S(1), \ldots, S(u)]$, of non-negative increasing integers satisfying $0 \leq S(1) < S(2) < \cdots < S(u) \leq m$. The third component is an u-tuple $[G(1), \ldots, G(u)]$ of positive increasing integers satisfying $1 \leq G(1) < G(2) < \cdots < G(u) \leq r$. If $G(u) = r$, then the structure is denoted as complete structure. The set of constraints (6) can be formulated for the structure $[u, S, G]$.

$$\sum_{i \in I} a_{ij}^{S(w)} y_i \geq G(w) \qquad \textit{for} \ \ j \in J, \ \ w = 1,\ldots,u \tag{6}$$

If a feasible solution $\boldsymbol{y}$ of the constraints (2), (5) and (6) exists for a complete $[u, S, G]$, then each user location j must lie at least in the radius $S(1)$ from $G(1)$ located service centers and in the radius $S(2)$ from $G(2) - G(1)$ additional service centers and so on up to the radius $S(u)$ from the $G(u) - G(u-1)$ service centers. The worst situated user perceives at most the generalized disutility given by (7).

$$H_{[u,S,G]} = S(1) \sum_{k=1}^{G(1)} q_k + \sum_{w=2}^{u} S(w) \sum_{k=G(w-1)+1}^{G(w)} q_k \tag{7}$$

An exposing structure is called valid if there is at least one feasible solution of the problem (2), (5) and (6). Then, the problem of min-max fair emergency system design is reduced to the problem of finding the valid exposing structure with minimal value. This surrogate problem was solved in [4] so that the lexicographically minimal valid structure was constructed by successive expanding an initial incomplete valid structure.

3 Searching Algorithm for the Optimal Exposing Structure

The searching algorithm presented below proceeds non-decreasing r-tuples of integers instead of exposing structures, which can be uniquely mapped on the set of exposing structures according to the rule (8).

$$\begin{aligned} m_k &= S(1) \qquad for\ \ k = 1, \ldots, G(1) \\ m_k &= S(w) \qquad for\ \ w = 2, \ldots, u,\ \ k = G(w-1)+1, \ldots, G(w) \end{aligned} \tag{8}$$

According to above-defined properties of the exposing structure, we will define valid k-tuple, incomplete r-tuple etc. The value $H_{[u,S,G]}$ defined by (7) for an complete exposing structure $[u, S, G]$ will be redefined as the value H_m of the associated r-tuple $\boldsymbol{m}$ according to (9).

$$H_{\boldsymbol{m}} = \sum_{k=1}^{r} q_k m_k \tag{9}$$

The searching algorithm is based on so-called improving step, which starts with input valid r-tuple $\underline{\boldsymbol{m}}$ and seeks for valid lexicographically minimal r-tuple $\boldsymbol{m}$, which is lexicographically greater than the r-tuple $\underline{\boldsymbol{m}}$ and fulfills $H_{\boldsymbol{m}} < H_{\underline{\boldsymbol{m}}}$. The search within improving step is performed by successive building up a valid k-tuple $m_1, \ldots, m_k$ with the minimal possible m_k in order to the value of the resulting r-tuple is less than the value of the r-tuple $\underline{\boldsymbol{m}}$. It can be easily proved that the choice of m_k for $k = 1, \ldots, r$ is limited by the inequalities (10) and (11) depending on k. The choice of m_k for $k = 1$ is subjected to (10).

$$\begin{aligned} \underline{m}_1 \le m_1 &< \left(\sum_{t=1}^{r} q_t \underline{m}_t \right) \Big/ \sum_{t=1}^{r} q_t \\ m_1 &< max \left\{ \left(\sum_{t=1}^{r} q_t \underline{m}_t - \underline{m}_{u-1} \sum_{t=u}^{r} q_t \right) \Big/ \sum_{t=1}^{u-1} q_t \ :\ u = 2, \ldots, r \right\} \end{aligned} \tag{10}$$

The choice of m_k for $k = 2, \ldots, r$ is limited by (11).

$$\begin{aligned} m_{k-1} \le m_k &< \left(\sum_{t=1}^{r} q_t \underline{m}_t - \sum_{t=1}^{k-1} q_t m_t \right) \Big/ \sum_{t=k}^{r} q_t \\ m_k &< max \left\{ \left(\sum_{t=1}^{r} q_t \underline{m}_t - \sum_{t=1}^{k-1} q_t m_t - \underline{m}_{u-1} \sum_{t=u}^{r} q_t \right) \Big/ \sum_{t=k}^{u-1} q_t \ :\ u = k+1, \ldots, r \right\} \end{aligned} \tag{11}$$

The algorithm of the improving step performs according to the following steps.

- Step 0. Initialize $k = 1$, $minM_1 = \underline{m}_1$.
- Step 1. Determine $maxM_k$ according to the associated upper limits of m_k in (10) and (11). Determine the lowest value of m_k from the range $minM_k, \ldots, maxM_k$ in order to the corresponding k-tuple $m_1, \ldots, m_k$ is valid. If m_k has been found, go to Step 2. Otherwise go to Step 3.
- Step 2. If $k = r$, terminate. The r-tuple $m_1, \ldots, m_r$ corresponds with the valid complete exposing structure, which has lower value than H_m. Otherwise update $k = k + 1$, $minM_k = m_{k-1}$ and go to Step 1.
- Step 3. If $k = 1$, terminate. No improving r-tuple has been found. Otherwise update $k = k - 1$, $minM_k = m_{k+1}$ and go to Step 1.

The complete searching process for the optimal exposing structure starts with an r-tuple, which corresponds to the lexicographically minimal valid exposing structure obtained according to [4]. This input r-tuple is used to initialize so-called current r-tuple $\underline{\boldsymbol{m}}$. The complete searching process consists of a cycle, in which the above-described improving step is repeatedly applied on the current r-tuple $\underline{\boldsymbol{m}}$. If an improved r-tuple $\underline{\boldsymbol{m}}$ is obtained, then the current r-tuple is updated by the improved r-tuple and the improving step is repeated. Otherwise, the algorithm terminates.

4 Numerical Experiments

The goal of this computational study is to explore the effectiveness of suggested algorithm as concerns the highest perceived disutility and the computational time as well. The proposed method was tested on the pool of benchmarks obtained from the road network of self-governing region of Košice. Several instances with a different number of located centers p were solved. The set of communities represents both the set J of users' locations and the set I of possible center locations. The experiments were performed for $r = 3$ and weight coefficients $q_1 = 1$, $q_2 = 0.2$, and $q_3 = 0.1$, which were preliminarily recommended by experts. To solve the problems described in previous sections, the optimization software FICO Xpress 7.9 (64-bit, release 2015) was used and the experiments were run on a PC equipped with the Intel Core i7 5500U processor with the parameters: 2.4 GHz and 16 GB RAM. The obtained results are reported in Table 1. Each row corresponds to one solved instance described by the value of p, which limits the number of centers to be located. The left section denoted by "BASIC STRUCTURE" is reserved for the basic approach, which finds lexicographically minimal exposing structure. The right part "ADVANCED STRUCTURE" contains the results of the suggested approach, which is able to find

Table 1 Results of numerical experiments for the self-governing region of Košice with $|I| = 460$ possible service center locations

p	Basic structure			Advanced structure		
	Time (s)	Structure S	H	Time (s)	Structure S	H
230	5	[4, 10, 14]	7.4	6	[4, 10, 14]	7.4
154	7	[5, 12, 16]	9.0	10	[5, 12, 16]	9.0
115	10	[7, 12, 16]	11.0	13	[7, 12, 16]	11.0
92	8	[7, 18, 25]	13.1	29	[8, 14, 19]	12.7
46	17	[12, 21, 27]	18.9	40	[12, 21, 27]	18.9
31	23	[15, 26, 43]	24.5	193	[15, 28, 35]	24.1
23	20	[18, 30, 43]	28.3	92	[18, 30, 43]	28.3
16	17	[21, 43, 55]	35.1	300	[22, 37, 51]	34.5
12	16	[25, 55, 64]	42.4	401	[26, 44, 60]	40.8
10	21	[28, 55, 70]	46.0	472	[29, 48, 65]	45.1
8	19	[32, 59, 79]	51.7	426	[32, 60, 72]	51.2

the optimal exposing structure with better value. For both approaches, three different results are reported: computational time in seconds, component S of the exposing structure and the value of H, which corresponds to the maximal disutility perceived by the worst situated users.

5 Conclusions

The main contribution of this paper consists in enhancing of the previously advanced approximate algorithm for the min-max location problem with generalized disutility, which is based on the radial formulation and exposing constraints. Whereas the approximate algorithm finds lexicographically minimal exposing structure, the suggested enhancement is able to find the optimal exposing structure with better value. Suggested approach was verified by series of numerical experiments performed with real data obtained from the road network of Slovakia. The results confirmed the efficiency of the associated algorithm. Thus, we considerably improved the useful tool for solving middle-sized min-max fair emergency system design problem.

Acknowledgements This work was supported by the research grants VEGA 1/0518/15 "Resilient rescue systems with uncertain accessibility of service", VEGA 1/0463/16 "Economically efficient charging infrastructure deployment for electric vehicles in smart cities and communities" and APVV-15-0179 "Reliability of emergency systems on infrastructure with uncertain functionality of critical elements".

References

1. Elloumi, S., Labbé, M., Pochet, Y.: A new formulation and resolution method for the p-center problem. INFORMS J. Comput. **16**, 84–94 (2004)
2. García, S., Labbé, M., Marín, A.: Solving large p-median problems with a radius formulation. INFORMS J. Comput. **23**(4), 546–556 (2011)
3. Janáček, J.: Radial approach to the emergency public service system design with generalized system utility. Int. J. Appl. Math. Inform. **8**, 7–14 (2014)
4. Janáček, J., Kvet, M.: Min-max optimization of emergency service system by exposing constraints. Communications: Scientific Letters of the University of Žilina vol. 2, pp. 15–22 (2015)
5. Janáček, J., Kvet, M.: Min-max optimization and the radial approach to the public service system design with generalized utility. Croatian Oper. Res. Rev. **7**(1), 49–61 (2016)
6. Marianov, V., Serra, D.: Location problems in the public sector. In: Drezner, Z. et al. (ed.) Facility Location: Applications and Theory, pp. 119–150. Springer, Berlin (2002)
7. Sayah, D., Irnich, S.: A new compact formulation for the discrete p-dispersion problem. Eur. J. Oper. Res. **256**(1), 62–67 (2016)

Solving a Rich Intra-facility Steel Slab Routing Problem

Biljana Roljic, Fabien Tricoire and Karl F. Doerner

Abstract We optimize the routing of steel slabs between locations in a steel production facility during a one hour-long operational period. Steel slabs are heterogeneous items that appear at locations at different release times. Certain slabs need to be delivered to another location before their specified due time. They are transported by fleets that include standard vehicles as well as truck-and-trailer type vehicles. The vehicles visit several locations multiple times. The input is such that not all slabs can be delivered in time, therefore two objective functions are provided that are organized in a lexicographic fashion: First, we maximize the throughput. Second, we aim to minimize travel times. An exact solution can only be obtained for small problem settings. In order to solve larger instances, we developed a heuristic. The results show that the solutions obtained by the heuristic reveal significant improvements to the real world solutions provided by our industrial partner.

1 Introduction

Increasing quality expectations, complex handling processes, and high throughput volumes lay particular stress on steel industries. The first product that occurs within the supply chain of steel production facilities is a steel slab. Steel slabs are heterogeneous items with specific handling instructions. According to those instructions, the cast slabs are cut, machined, stored at open air fields to cool off, stored in warm-holding boxes to remain at a certain temperature, or brought straight to the rolling mill. All those production stages are carried out at different locations inside the factory. Eventually, all steel slabs end up at the rolling mill where they are transformed into coils. In our problem setting, the production and the rolling schedules as well as the required handling stages for each steel slab are predetermined. Furthermore, a

B. Roljic (✉) · F. Tricoire · K.F. Doerner
Department of Business Administration, University of Vienna,
Oskar-Morgenstern-Platz 1, 1090 Vienna, Austria
e-mail: biljana.roljic@univie.ac.at

F. Tricoire
e-mail: fabien.tricoire@univie.ac.at

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_42

fleet of standard vehicles and a fleet of truck-and-trailer type vehicles are given. The latter consists of trucks that serve as towing vehicles and cannot hold items. Each truck can pull a single trailer and a trailer cannot move without a truck. The number of available trailers always exceeds the number of available trucks. The scope of our optimization problem is the routing of steel slabs along their production stages in such a way that, primarily, the throughput expectations are met while, secondarily, the travel time dependent logistics costs are kept low.

Meisel and Kopfer [2] have already contributed to similar problem settings, namely the Active-Passive Vehicle Routing Problem (APVRP), where a truck pulls a trailer and a trailer is used for holding cargo. Our work additionally takes into account that both the standard vehicle and the trailer can fulfill multiple transportation requests simultaneously, hence transport several slabs at once up to a maximum total weight and total number. Tilk et al. [5] presented a branch-and-price-and-cut algorithm for the exact solution of the APVRP. According to the categorization by Drexl [1], our work investigates the so-called *movement synchronization en route*; that is, the active and passive means of transport have to traverse the same arc at the same time, and joining and separating vehicles is possible at any location that they visit during their routes. Among the Pickup and Delivery literature, our routing problem can be assigned to the problem class of Vehicle Routing Problems with Pickups and Deliveries (VRPPD) with paired pickup and delivery points as a part of the less-than-truck-load problems, as categorized by Parragh et al. [3].

In the real world setting, incompatibilities between vehicles and steel slabs apply. The real world results of our industrial partner on the standard vehicle routing and truck-and-trailer routing are treated separately. For better comparison of our solutions to those of our industrial partner, we will conform to this separation and handle the routing of the two fleets individually.

2 Problem Description

A transportation request $r \in R$ is defined for each steel slab that needs to be moved from one node to another during the considered period. The request includes a pickup location α^r and a delivery location β^r, a release time s^r formulated as a lower bound on the pickup time, and a due time d^r formulated as an upper bound on the delivery time. We define a fleet K. The fleet size is predetermined. Standard vehicles are specified in $S \subseteq K$, whereas trucks (active means of transport) and trailers (passive means of transport) are denoted by $A \subseteq K$ and $P \subseteq K$ respectively. Furthermore, the vehicles in $S \subseteq K$ and $P \subseteq K$ have a capacity in terms of weight Q^k as well as number of items that can be simultaneously transported U^k. Decoupling a trailer from a truck is very quick, but loading and unloading items takes time σ^{rk}. We are considering an undirected graph $G = (V, A)$ where V is the set of nodes and A is the set of arcs. The nodes represent real world locations that are visited by vehicles performing pickups and/or deliveries of one or multiple steel slabs at once. Since transportation requests may define the same pickup and/or delivery location, we allow vehicles

to visit nodes multiple times. Each arc $(i, j) \in A$ is associated with a travel time t_{ij}^{k} that differs between vehicles. The objective function is organized in a lexicographic fashion. First, maximize the throughput, then, minimize the total travel time of the fleet.

We keep track of each vehicle's legs, which are denoted with index l, resulting in the vehicle flow variables x_{ij}^{kl}. Similarly, we keep track of each item leg, denoted using index h, since other intermediate locations can be visited between an item's pickup and delivery location. Thus, we introduce the request flow variables w_{ij}^{rh}. The flow variables x_{ij}^{kl} and w_{ij}^{rh} need to be synchronized, which is done using the flow variables y_{ij}^{klrh}.

The complete mathematical model grows quickly in size and exact solutions can only be obtained for small instances of up to 10 locations and 20 requests. The amount of vehicle and item legs may be parametrized to decrease the computational complexity. Reducing the number of legs, allows us to receive exact results for bigger instances, though, due to the sacrifice in legs, we are restricted to explore a reduced solution space. Therefore, a heuristic solution approach is required for solving real world instances in a reasonable amount of computational time.

3 Solution Approach

To solve our routing problem for standard vehicles and trailers, we develop an insertion heuristic that constructs a route based on pickup-visits and delivery-visits of requests, rather than pickup nodes and delivery nodes. Every transportation request holds a pair of visits that should be inserted during the procedure of the heuristic. A pickup visit of request $r \in R$ is denoted by v_r^+, while a delivery visit is denoted by v_r^-. Both types of visits hold information on the corresponding node, time window, and weight of the requested steel slab. For every request $r \in R$ we evaluate the insertion of their pickup-visit v_r^+ and delivery-visit v_r^- at every possible pickup position p and every possible delivery position d into all routes $t \in T$. The evaluation of the insertion is done by a cheapest insertion rule that compares the resulting total travel times of the routes when feasibly inserting a request. A candidate list C is introduced that is structured so that for every request $r \in R$, it holds information on which routes $t \in T$ can be selected for a feasible insertion. For each of those routes it remembers at which positions p and d the insertion of the pickup-visit v_r^+ and delivery-visit v_r^- of the considered request would be the cheapest. In every iteration, the candidate list is searched for the cheapest insertion among all requests it contains. The solution is constructed by inserting the chosen request r_{best} into the corresponding route t_{best}. After this step, the insertion costs of the remaining requests are updated by re-evaluating the insertion of their pickup-visit v_r^+ and delivery-visit v_r^- in the modified route. This procedure is terminated once all requests are inserted or once all remaining requests in candidate list C cannot be feasibly inserted in any route.

INSERTION HEURISTIC (IH): STANDARD VEHICLE AND TRAILER ROUTING (solution s)

for each request $r \in R$ **do**
    determine for every feasible route $t \in T$ the best pickup position p for inserting pickup-visit v_r^+ and the best delivery position d for inserting delivery-visit v_r^- according to a cheapest insertion rule (post insertion smallest total travel time c_t of route t), and save it to the candidate list C (candidate list C is structured by requests $r \in R$);
end for
while candidate list C contains feasible insertion options **do**
    $c_{t,best} \leftarrow M$;
    for each request $r \in C$ **do**
        determine route t_b with the smallest total travel time $c_{t,b}$ post insertion of request r;
        if $c_{t,b} < c_{t,best}$ **then**
            $r_{best} \leftarrow r$;
            $t_{best} \leftarrow t_b$;
            $c_{t,best} \leftarrow c_{t,b}$;
        end if
    end for
    insert pickup-visit v_r^+ and delivery-visit v_r^- of request r_{best} at positions p and d into t_{best};
    update solution s by route t_{best};
    erase request r_{best} from candidate list C;
    for every request $r \in C$ holding t_{best} as an insertion option, re-evaluate the best positions p and d for inserting the visit-pair v_r^+ and v_r^-, and update it to candidate list C;
end while
return solution s

3.1 Large Neighbourhood Search

Large Neighbourhood Search (LNS) is a meta-heuristic that was introduced for VRPTW by Shaw [4]. In LNS, an initial solution is gradually improved by alternately destroying and repairing the solution. We apply LNS for improving our initial solution constructed by the previously described insertion heuristic IH.

We copy solution s to solution s_{new}. In order to select item requests to destroy from solution s_{new}, we use a roulette wheel with the number of item legs as weights. The pickup-visits and delivery-visits of the selected item requests are removed from solution s_{new}. Then, a repair operator, specifically the insertion heuristic IH, reinserts the unfulfilled item requests into solution s_{new}. Solution s_{new} is accepted only if the throughput of s_{new} is bigger than the throughput of s. If the throughput of s_{new} is equal to the throughput of s, then the solution with the shorter travel time is favoured.

3.2 Truck Routing

In this section, we assign trucks to route trailers. Note that we always have less trucks than trailers available. Thus, it is valid to expect that not all trailer legs can be feasibly routed in the truck solution s_{trucks}. First, the trailer routing is solved by the insertion heuristic IH and improved by LNS. The solution is represented in solution

$s_{trailers}$. In order to solve the truck routing, we transform the trailer routes t included in solution $s_{trailers}$ into multiple truck transportation requests $\omega \in \Omega$. Specifically, every trailer leg represents a transportation request defining pickup and delivery nodes as well as a corresponding time window for performing the transportation of the trailer as soon as it is loaded. The truck capacity U^k is set to 1, since a truck can only pull one trailer. Then, we can apply the insertion heuristic IH for the truck routing. Note that transportation requests, representing trailer legs, comprise interrelated and overlapping slab requests. For example, a slab might be picked up at the beginning of a certain trailer leg, but delivered at the end of a chronologically much later leg in a trailer's route. This slab would span multiple item legs. In order to fulfil the transportation of such an item, all trailer legs it uses need to be included in the truck solution s_{trucks}. Because of this, we need to verify the interdependencies of the transportation requests. This is done by examining which items are being transported on the considered transportation request ω. Based on this information, we can retrace the preliminary and/or postliminary dependencies to neighbouring transportation requests. If such dependencies exist, we impose that all interrelated ω must be feasibly included in solution s_{trucks}. If a feasible insertion of the combined ω cannot be obtained, then we continue to another transportation request and again repeat the verification process on the dependencies. In case that multiple transportation requests compete for a truck during the same time span, then the transportation request that involves a larger number of fulfilled item requests will be favoured.

4 Results

Table 1 shows the computational results of the steel slab routing performed by standard vehicles. An initial solution is generated by the insertion heuristic IH and further improved by LNS. For LNS, we define a destruction rate of 30% and a stopping criterion that terminates the LNS when after 20 iterations no improved solution can be found. The real world results show the maximum throughput that our industrial partner met in the past. The gap expresses the decrease or increase of travel time t as a percentage when comparing the results of the real world case RW, solution approach IH, and solution approach LNS with one another. We can see that IH produces solutions that are significantly better than the real world solutions in six cases out of eight, and worse in the two remaining cases. However, LNS performs better than both IH and the real world solutions in all cases.

5 Summary and Outlook

We presented a solution approach for solving rich intra-facility steel slab routing problems, including many specifics from real world cases. Our results are competitive to those provided by our industrial partner, and in a large number of cases

Table 1 Computational results for standard vehicle routing

Real world (RW)					Insertion/Construction heuristic (IH)				Large neighbourhood search (LNS)					
Inst	$\|R\|$	$\|S\|$	d	t	d	t	cpu	$gap^{t}_{RW,IH}$ (%)	d	t	$\#it$	cpu	$gap^{t}_{RW,LNS}$ (%)	$gap^{t}_{IH,LNS}$ (%)
i_1	25	3	25	55	25	35	0.3	36.4	25	32	41	5.1	41.8	8.6
i_2	25	3	25	43	25	49	0.5	–14.0	25	35	23	3.1	18.6	28.6
i_3	37	5	37	127	37	88	0.9	30.7	37	84	41	14.5	33.9	4.5
i_4	37	5	37	84	37	97	0.9	–15.5	37	83	34	11.9	1.2	14.4
i_5	67	4	67	105	66	80	8.0	23.8	67	75	45	109.1	28.6	6.3
i_6	67	4	67	99	66	86	9.1	13.1	67	75	62	151.4	24.2	12.8
i_7	100	5	100	256	95	242	13.4	5.5	100	215	56	262.2	16.0	11.2
i_8	100	5	100	196	91	173	12.7	11.7	100	162	47	217.6	17.3	6.4

instances $i_7 - i_8$ correspond to a 2 h-long operational period, throughput d in pieces, travel time t in minutes, cpu in seconds, $\#it$ number of iterations

significantly better. Specifically, the results of the LNS always meet the throughput of the real world case while decreasing the travel time. We plan to extend our solution approach to optimize larger instances that include up to 1000 requests as well as an extended objective function considering fleet minimization. Further, we want to consider partial deliveries by multiple vehicles such that the transportation of an item can be carried out by different vehicles.

Acknowledgements This research is funded by the Austrian Research Promotion Agency (FFG) through K-Project #843532 HOPL. The authors would like to thank Voestalpine AG and Logistik Service GmbH for providing detailed insights.

References

1. Drexl, M.: Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints. Transp. Sci. **46**(3), 297–316 (2012)
2. Meisel, F., Kopfer, H.: Synchronized routing of active and passive means of transport. OR Spectr. **36**(2), 297–322 (2014)
3. Parragh, S.N., Doerner, K.F., Hartl, R.F.: A survey on pickup and delivery problems Part II: transportation between pickup and delivery locations. J. fuer Betriebswirtschaft **58**, 81–117 (2008)
4. Shaw, P.: New Local Search Algorithm Providing High Quality Solutions to Vehicle Routing Problems. Technical report, APES Group, Department of Computer Science, University of Strathclyde, Scotland (2012)
5. Tilk, C., Bianchessi, N., Drexl, M., Irnich, S., Meisel, F.: Branch-and-Price-and-Cut for the Active-Passive Vehicle-Routing Problem, forthcoming in Transportation Science

Splitting Procedure of Genetic Algorithm for Column Generation to Solve a Vehicle Routing Problem

Martin Scheffler, Christina Hermann and Mathias Kasper

Abstract This paper considers an extended Vehicle Routing Problem with Simultaneous Pickup and Delivery and Time Windows (VRPSPDTW). For this problem we describe a simple but effective extension of a genetic algorithm (GA) based on chromosome permutation and a splitting procedure. For large instances it is obvious to use the original GA as benchmark. These approaches are applied on test instances for the considered problem and on Solomon instances.

1 Introduction

Demographic change and the associated increase in the average age of the population results in a growing importance of home healthcare logistics. Therefore, the efficient design of routes under a variety of additional constraints is necessary.

We consider a Vehicle Routing Problem with Simultaneous Pickup and Delivery and Time Windows (VRPSPDTW) extended by two kinds of demands assigned to specific nodes, introduced by [8]. The first demand concerns delivery from specific node one, e.g. a drugstore s to customers (patients) i and the second concerns pickup from customers to specific node two, e.g. a lab l. Examples are the delivery of bandaging material from depot 0 to the customers, meds from the drugstore

M. Scheffler (✉)
Technische Universität Dresden, Fakultät Wirtschaftswissenschaften,
Lehrstuhl für BWL, insb. Industrielles Management, 01062 Dresden, Germany
e-mail: martin.scheffler@tu-dresden.de

C. Hermann · M. Kasper
Technische Universität Dresden, Fakultät Verkehrswissenschaften "Friedrich List",
Professur für Verkehrsbetriebslehre und Logistik, 01062 Dresden, Germany
e-mail: christina.hermann@tu-dresden.de

M. Kasper
e-mail: mathias.kasper@tu-dresden.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_43

to the customers and the collection of medical waste from customers to the depot respectively blood samples from the customers to the lab.

On the one hand, there is an abundance of studies about tuning common heuristics and using them for similar problems without a detailed view on the utilization of the generated information. For a recent review on common heuristics, see [1] or [6]. On the other hand, there are many studies about column generation concerning the Vehicle Routing Problem (VRP), e.g. [2, 5].

The underlying concept of this approach is extending a stand-alone heuristic (searching for a set of routes) by column generation. Therefore, this paper describes an extension of a genetic algorithm (GA) based on chromosome permutation and a splitting procedure introduced by [10]. The main idea of the hybrid algorithm (HA) is using the splitting procedure to create a complete memory of all generated valid trips as feasible columns. This enables us to solve the set partitioning problem (SPP). Instead of using the SPP only as post optimization, the SPP is solved several times during runtime. Due to the resulting interactions, both parts (GA, SPP) can influence each other (positively). For practical application, there is a need of fast generated (solution time <10 min) and feasible solutions. Hence, in a first step we have used simple genetic operators and a relatively small number of iterations.

To the best of our knowledge, there are no similar algorithms. Therefore, in Sect. 2 we describe the rudimentary GA with a detailed view on the splitting procedure for the considered problem. Section 3 presents the extended use of the splitting procedure and the resulting HA. Section 4 reports the results of our computational experiments. These approaches are tested on instances for the considered problem and by adjusted input data on Solomon instances [11].

2 Genetic Algorithm

The GA is based on encoding a solution as chromosome by a permutation (sequence) of all customers and on a splitting procedure to extract the optimal solution of this sequence [10]. The population size is set to 30 individuals. The fitness of an individual is the total cost of the resulting VRP solution. As genetic operator a One-Point-Crossover is only used. We select the first parent randomly from the better half of the sorted population. The second parent is selected from the worse half. The crossover produces two children which are evaluated by the splitting procedure. After each iteration the population is sorted in ascending order by the fitness value. Taking account of a practicable computation time (<10 min), the maximum number of iterations is set to 60.000 and after every 3.000 iterations a variation is carried out by replacing the worse half of the population with random chromosomes. Clearly, this is the simplest version of a GA, but for a better comparison, considering the special characteristics of the HA (see Sect. 3), and the use of the GA as benchmark for the HA (see Sect. 4), it is sufficient. The initial solution is created by the Savings-heuristic, see [3]. The Savings-list is created as shown in [8]. An initial population is built by random permutations of routes (initial solution).

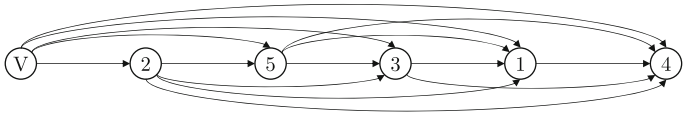


Fig. 1 Auxiliary graph H

Table 1 Possible routes for arc (2, 4)

0	*s*	5	3	*l*	1	4	0
0	*s*	5	3	1	*l*	4	0
0	*s*	5	3	1	4	*l*	0
0	5	*s*	3	*l*	1	4	0
0	5	*s*	3	1	*l*	4	0
0	5	*s*	3	1	4	*l*	0

Table 2 Example assumptions

i	0	1	2	3	4	5	s	l
d_i^s	0	1	1	2	4	0	0	0
p_i^l	0	0	3	1	0	2	0	0

Since the HA is based on an extension of the splitting procedure, it is important to review the directed auxiliary graph H. Figure 1 shows the complete graph for an example with five costumers. Each arc represents a feasible route for serving the included costumers in the order given by the chromosome.

Therefore, a real auxiliary graph is not complete. For example arc (2, 4) represents such a route for customers 5, 3, 1, 4. Node V is appropriate for a uniform interpretation of an arc: including all nodes, excluding first node, including last node. For the considered problem an arc of the graph additionally represents the route with minimum costs. Depending on the demand of each customer i concerning s (d_i^s) or l (p_i^l), there is more than one possible route for each arc. Table 1 shows all possible routes of arc (2, 4) considering the assumptions shown in Table 2.

The optimal solution for the given sequence of customers is determined by solving a restricted shortest path problem given by H applying dynamic programming considering the number of vehicles. Each arc is weighted equal to the tour cost.

3 Hybrid Algorithm

The HA creates a complete memory of all feasible routes generated in each iteration. That means all feasible routes of graph H are saved regardless of offspring admissi-

bility and iteration productivity. Saving clones of routes is prohibited. Therefore we are able to solve the SPP periodically.

Note that, there is a trade-off between finding a better solution with the GA and creating new unique columns for the SPP. For this reason the GA is designed straightforward. For additional genetic operators the impact on both heuristics has to be tested. Furthermore it is conceivable to design special operators to create only unique routes. The variation rather than frequency is also a useful instrument to create unique routes.

The special aspect of the HA is that any improvement of the GA is also an improvement of that. However, the memory management and solving the SPP need extra time. Therefore, it is appropriate to set a time limit for the HA. Obviously, the computing time of the GA is used. This leads to the question of frequency of solving the SPP. Hence, we tested several common settings for the frequency with 60.000 iterations. Note that, the frequency is synchronised with the variation. Thus, it was a test for both issues.

The integration of the SPP-solution into population and its variation are done parallel. Out of the SPP-solution 10 chromosomes and 10 random chromosomes are created. For a first evaluation the GA was applied with identical settings. Each heuristic was tested three times on each instance (50 customers).

The results of the HA are better than those of the GA (costs of best solution, average costs of solution). The GA results are nearly identical for each setting. An important fact is that the GA creates less feasible solutions than the HA. The selected setting of the test is a variation (and solving the SPP) every 3.000 iterations.

4 Computational Results

For the computational approaches own test instances are generated. The instances are created by generating spatial data, time data and demand data separated from each other and combining them afterwards. 3 sets of spatial data with random (R), clustered (C) and random clustered (RC) distribution are generated. Time windows are generated randomly considering an eight-hour day (d, s, l). There is a narrow set (n, 30–90 min) and a wide set (w, 30–210 min) of time windows. Each kind of demand is also generated randomly. A set is generated under the assumption that 50% (m) of the customers need between 1 and 5 goods of each demand (each other customer 0 goods). Similarly, a set is generated with 80% (h). Combining spatial, time and demand data results in 12 instances. These are created for 50 (only Sect. 3) and 100 customers. The data are available on request. Each heuristic is tested three times on each instance.

Each algorithm is implemented in C# and is run on an Intel(R) Xeon(R) CPU E5-2630 v2 @ 2.60GHz server with 12 cores and 384 GB RAM. The SPP is solved with Gurobi 6.0. Note that, the implementation is based on single core use only and for Gurobi a limit was set to 6 cores. The used memory does not exceed 1 GB (HA).

The evaluation is carried out using two different methods. First, the GA is used as benchmark for the HA. Second, for the Solomon instances [11] both heuristics are compared with the optimal upper bound (S_{UB}) summarized by [7] and with the results of the memetic algorithm (NDB10) of [9], one of the best state of the art metaheuristic, shown by [4]. Table 3 shows the results of the considered problem. Bold entries mark the best value in each category.

The HA outperforms the GA regarding costs of best identified solution and average costs of all solutions. In case of small solution space the HA is more capable to create feasible solutions. The standard deviation of the GA is slightly better than the obtained value of the HA, but it is easier to create constant bad than good solutions.

Both heuristics are tested on the Solomon-instances with adjusted input data ($d_i^s = p_i^l = p_i^0 = 0$). Table 4 shows the results of the Solomon instances. The performance of the HA is better than that of the GA on the Solomon instances, too. The GA performs better for instances with large solution spaces (C201-C208). It can be assumed that this effect can be reduced with additional genetic operators. The memetic algorithm outperforms the HA on each Solomon instance. For easy comparison, it is advisable to use the cumulated costs of the best solution of all instances *CCB*. Because the GA does not create feasible solutions for each instance, it is disregarded. The CCB_{NDB10} is 57.187 with an average computing time of 16, 9 min per instance. The HA produces a CCB_{HA} of 63.205 in an average computing time (8, 2 min) less than the half of NDB10. This corresponds to a deviation of 9, 5% between HA and NDB10. By disregarding C201-C208 the deviation falls to 5, 4%. Given a much lower computing time and the use of a very simple genetic operator, this is assessed as well.

5 Conclusions and Further Research

In this paper we present a new hybrid algorithm. We extend a common GA by column generation. Columns are generated by the splitting procedure, both in the GA and HA. The extension is using this columns across iterations by saving them and solving the SPP. The results show that the HA outperforms the GA under identical settings. For instances with large solution space it was ascertained that additional and more effective genetic operators are necessary for the HA. Nevertheless, it was shown that the basic concept operates well. So there is a huge potential of the HA.

For further research we have to check a state of the art GA as a basis for the HA. Furthermore, changing the SPP to a set covering problem (SCP) in addition with a repair procedure to remove duplicate customers from the SCP solution has to be tested. Perhaps, an improvement of the SCP solution can be achieved. Moreover, the utilization of the dual variables (in combination with solving the relaxed SPP/SCP) for this kind of GA has to be investigated. Finally, the authors will continue the research. The comparison of the results of small instances to the optimal solution of a MIP-formulation is essential.

Table 3 Results instances 100 customers

Instances	CPU	GA_b	HA_b	GA_a	HA_a	GA_s^r (%)	HA_s^r (%)	$HA_a < GA_a$ (%)	NoV	Q
R_100_w_m	287	345	**296**	348	**308**	**0, 8**	2, 9	17, 9	20	50
R_100_w_h	185	475	**342**	476	**350**	**0, 3**	2, 4	39, 3	20	50
R_100_n_m	82	–	322	–	326	–	1, 0	–	20	50
R_100_n_h	204	480	**362**	490	**381**	**2, 2**	4, 8	35, 2	20	50
C_100_w_m	483	272	**229**	285	**230**	3, 4	**0, 6**	24, 1	20	50
C_100_w_h	368	410	**287**	419	**295**	**1, 7**	2, 1	46, 0	20	50
C_100_n_m	185	–	237	–	243	–	1, 9	–	20	50
C_100_n_h	202	437	**299**	451	**304**	2, 4	**1, 5**	50, 6	20	50
RC_100_w_m	306	303	**253**	307	**258**	1, 4	**1, 3**	21, 2	20	50
RC_100_w_h	211	409	**319**	415	**333**	**1, 7**	3, 0	30, 1	20	50
RC_100_n_m	491	332	**278**	345	**286**	3, 7	**3, 6**	24, 0	20	50
RC_100_n_h	384	440	**328**	453	**341**	**1, 9**	2, 6	37, 9	20	50

Notes Notation of instances: Distribution_NumberOfCustomers_TimeWindows_Demand. *CPU* computing time (seconds). GA_b costs of the best solution identified by GA (km). HA_b costs of the best solution identified by HA (km). GA_a average costs of all solutions created by GA (km). HA_a average costs of all solutions created by HA (km). GA_s^r relatively standard deviation between *GA* and GA_a. HA_s^r relatively standard deviation between *HA* and HA_a. -: no solution found, *NoV* number of vehicles, *Q* vehicle capacity

Table 4 Results solomon instances 100 customers

R	S_{UB}	NDB10$_b$	GA_b	HA_b	C	S_{UB}	NDB10$_b$	GA_b	HA_b	RC	S_{UB}	NDB10$_b$	GA_b	HA_b
101	1638	1651	–	**1736**	101	827	829	928	928	101	1620	1697	–	**1828**
102	1467	1486	1728	**1629**	102	827	829	971	**945**	102	1457	1555	1885	**1717**
103	1209	1293	1455	**1392**	103	826	828	1027	**927**	103	1258	1262	1559	**1518**
104	972	1007	1111	**1108**	104	823	825	945	**921**	104	1132	1136	1210	**1204**
105	1355	1377	1613	**1582**	105	827	829	930	**928**	105	1514	1629	1909	**1822**
106	1235	1252	1446	**1405**	106	827	829	**943**	944	106	1373	1425	1601	**1590**
107	1065	1105	1272	**1234**	107	827	829	928	**927**	107	1208	1231	1349	**1326**
108	932	961	1078	**1059**	108	827	829	847	847	108	1114	1140	1257	1257
109	1147	1195	1335	**1282**	109	827	829	**843**	847					
110	1068	1119	1201	**1180**										
111	1049	1097	1196	**1183**										
112	949	982	1051	**1044**										
201	1143	1252	1427	**1359**	201	590	592	**592**	752	201	1262	1407	1746	**1600**
202	1030	1192	1253	**1192**	202	589	592	**716**	785	202	1092	1366	1519	**1406**
203	871	940	1045	**1031**	203	589	592	810	**760**	203	924	1050	1179	**1154**
204	–	826	**839**	841	204	588	591	**765**	784	204	–	799	**901**	942
205	–	994	1095	**1082**	205	586	589	**633**	643	205	1154	1298	1536	**1464**
206	–	906	1066	**1063**	206	586	589	**643**	662	206	1051	1146	1477	**1438**
207	–	891	947	**916**	207	586	588	**656**	664	207	–	1061	1221	**1152**
208	–	727	**794**	798	208	586	588	**679**	679	208	–	828	**858**	862
209	855	909	981	**960**										
210	–	939	1043	**1023**										
211	–	886	898	**887**										

Notes For CPU, GA_b and HA_b see Table 3. S_{UB} upper bound shown by [7]. $NDB10_b$ costs of the best solution identified by the memetic algorithm of [9]. -: no solution found or no existing optimal upper bound

References

1. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows. Part II: metaheuristics. Transp. Sci. **39**(1), 119–139 (2005)
2. Chabrier, A.: Vehicle routing problem with elementary shortest path based column generation. Comput. Oper. Res. **33**(10), 2972–2990 (2006). Part Special Issue: Constraint Programming
3. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**(4), 568–581 (1964)
4. Desaulniers, G., Madsen, O.B., Ropke, S.: The vehicle routing problem with time windows. In: Toth, P., Vigo, D. (eds.) Vehicle Routing: Problems, Methods, and Applications, chapter 5, pp. 119–159. SIAM, 2 edition
5. Desrochers, M., Desrosiers, J., Solomon, M.: A new optimization algorithm for the vehicle routing problem with time windows. Oper. Res. **40**(2), 342–354 (1992)
6. Gendreau, M., Potvin, J.-Y., Bräumlaysy, O., Hasle, G., Løkketangen, A.: Metaheuristics for the vehicle routing problem and its extensions: a categorized bibliography. In: The Vehicle Routing Problem: Latest Advances and New Challenges, pp. 143–169. Springer (2008)
7. Jepsen, M., Petersen, B., Spoorendonk, S., Pisinger, D.: Subset-row inequalities applied to the vehicle-routing problem with time windows. Oper. Res. **56**(2), 497–511 (2008)
8. Liu, R., Xie, X., Augusto, V., Rodrigez, C.: Heuristic algorithm for a vehicle routing problem with simultaneous delivery and pickup and time windows in home health care. Eur. J. Oper. Res. **230**, 475–486 (2013)
9. Nagata, Y., Bräysy, O., Dullaert, W.: A penalty-based edge assembly memetic algorithm for the vehicle routing problem with time windows. Comput. Oper. Res. **37**(4), 724–737 (2010)
10. Prins, C.: A simple and effective evolutionary algorithm for the vehicle routing problem. Comput. Oper. Res. **31**(12), 1985–2002 (2004)
11. Solomon, M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. **35**(2), 254–265 (1987)

Request-Allocation in Dynamic Collaborative Transportation Planning Problems

Kristian Schopka and Herbert Kopfer

Abstract Several publications on collaborative transportation planning problems (CTPPs) focus on schemes that ensure a fair assignment of collaborative profits. However, it is seldom taken into account that an even allocation of transportation resources (e.g. transportation requests) is also responsible for the viability and stability of horizontal carrier coalitions; particularly if dynamic CTPPs are considered. In this paper, the winner determination problem (WDP) of an auction-based request exchange is restricted by lower and upper bounds that respect an equality between transferred and received requests for carriers. In a computational study, the restricted WDP is applied to the dynamic collaborative traveling salesman problem.

1 Introduction

In dynamic transportation planning problems (TPPs), small and medium sized carriers (SMCs) are confronted with customers demanding for quick fulfillment of (transportation) requests. It means, new requests (referred to as incoming requests) appear during a planning period and have to be dispatched in the same period [2]. To overcome the uncertainty associated with incoming requests, rivaling SMCs ally in horizontal coalitions for auction-based request exchanges (ABREs). Thereby, requests are reallocated based on bids (maximal willingness to pay for request transfer). ABREs are able to reduce the transportation costs up to 15% [4]. To ensure an even assignment of collaborative profits, game theoretical schemes like the Shapley value [5] are integrated in ABREs. However, those schemes do not respect shifts in the request-portfolios of the individual SMCs that occur by repeatedly executed ABREs; e.g. within a rolling horizon planning (cf. [6]). Over time, an uneven allocation of

K. Schopka (✉) · H. Kopfer
Chair of Logistics, University of Bremen, Wilhelm-Herbst-Str. 5,
28359 Bremen, Germany
e-mail: schopka@uni-bremen.de
URL: http://www.logistik.uni-bremen.de

H. Kopfer
e-mail: kopfer@uni-bremen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_44

requests causes power-shifts among formerly commensurate and equal SMCs and may decrease the stability within horizontal coalitions.

In this paper, the winner determination problem (WDP) of an ABRE is restricted by lower and upper bounds that limit the number of reallocated requests. It means that all agents (e.g. traveling salesmen) transfer and receive a proportional number of incoming requests through an ABRE over a planning period. The effect of the restricted WDP is analyzed in a computational study on the dynamic collaborative traveling salesman problem (DCTSP). A mathematical formulation of the DCTSP is given in Sect. 2. Section 3 introduces a two-stage solution framework (TSF) for the DCTSP. While the restricted WDP and the agents' specific TPPs are solved by a mathematical solver, the calculation of bids is executed by a cheapest insertion algorithm. The findings of the computational study are presented in Sect. 4.

2 Problem Description

Let us consider a horizontal coalition among a set of rivaling agents $P = \{1, 2, \dots, |P|\}$. At the start of a planning period T, each agent is in charge of a set of requests N_p^s. The aim of agent $p \in P$ is to find the round trip that dispatches each request $i \in N_p^s$ once, starts and ends at the own depot O_p, and minimizes the transportation costs. Over time, each agent $p \in P$ receives a set of incoming requests N_p^c. All incoming requests have to be dispatched during T. To reduce transportation costs, the agents use an ABRE for the reallocation of incoming requests. It means, an incoming request $i \in N_p^c$ can either be served by the round trip of agent p or transferred to another agent $p^* \in P \setminus \{p\}$ within the coalition. The DCTSP can be split in agents' specific TPPs (Eqs. (1)–(7)) and a request reallocation problem (Eqs. (8)–(9)).

$$max \quad z(p) = \sum_{i \in N_p} \sum_{j \in N_p} (e_j - c_{ij}) \cdot x_{ij}^p, \tag{1}$$

$$s.t. \quad \sum_{i \in N_p} x_{ij}^p = 1, \qquad \forall j \in N_p^s \cup O_p, \tag{2}$$

$$\sum_{i \in N_p} x_{ij}^p \leq 1, \qquad \forall j \in N^c, \tag{3}$$

$$\sum_{j \in N_p} x_{ij}^p = \sum_{j \in N_p} x_{ji}^p = v_i^p, \qquad \forall i \in N_p, \tag{4}$$

$$\sum_{i,j \in S} x_{ij}^p \leq \sum_{j \in S \setminus \{k\}} v_j^p, \qquad \forall S \subset N_p \setminus \{0\}, k \in S, \tag{5}$$

$$x_{ij}^p \in \{0, 1\} \qquad \forall i, j \in N_p \times N_p, \tag{6}$$

$$v_i^p \in \{0, 1\} \qquad \forall i \in N_p. \tag{7}$$

The TPP relates to the traveling salesman problem with profits (cf. [1]) that identifies the most profitable round trip on the graph $G_p = (N_p, A_p)$ for each agent $p \in P$. While A_p is the edge set, $N_p := O_p \cup N_p \cup N^c$ builds the node set; let $N^c := \sum_{p \in P} N_p^c$ be the set union of incoming requests of all agents $p \in P$. The usage of an edge $(i,j) \in A_p$ requires transportation costs c_{ij}. The binary decision variable x_{ij}^p is equal to one if the edge $(i,j) \in A_p$ is included in the round trip of agent p, and it is zero, otherwise. Separately, the binary decision variable $v_i^p = 1$ shows that a request i is dispatched by agent p. Since each request j generates a freight rate e_j when it is dispatched, the Objective (1) maximizes the profits $z(p)$ (i.e. freight rates minus transportation costs). Constraints (2) and (3) ensure that each node is dispatched at most once. Constraints (4) observe the flow and sets v_i^p. Constraints (5) exclude sub-cycles (cf. [1]). Constraints (6)–(7) define the domains of the decision variables.

The request reallocation problem observes the exchange of incoming requests. Objective (8) identifies the combination of agents' specific round trips that maximize the overall collaborative profits $z(P)$, by respecting that each incoming request has to be dispatched by one agent of the coalition (Constraints (9)).

$$max \qquad z(P) = \sum_{p \in P} z(p), \tag{8}$$

$$s.t. \qquad \sum_{p \in P} \sum_{i \in N_p} x_{ij}^p = 1, \qquad \forall j \in N^c. \tag{9}$$

Due to the dynamic scenario of incoming requests, not all planning relevant data are known at the beginning of the planning period. To consider this issue, we supplement the previously presented mathematical models by a time factor. That is why the following features of dynamic planning have to be respected by computing the mathematical models: (i) an incoming request cannot be reallocated/dispatched before it is known; (ii) each request is deleted from the request pools after it is dispatched; (iii) the start positions of the round trips have to be updated.

3 Solution Methodology

To solve the DCTSP, we developed a TSF that organizes the ABRE by a combinatorial auction (CA). The CA can be split in agents' specific bid generations and a common WDP. To retain the independent decision power for all agents, the planning of the round trips and the bid generation are executed by separate planning steps that each agent gets through in an isolated planning. Against this situation, the WDP is solved by a mediator. Furthermore, our TSF suggests that the mediator assigns the collaborative profits by the Shapley value [5] among the agents.

In the first stage of TSF, for each agent $p \in P$ a round trip of all requests in N_p^s is generated. Therefore, the TPP is computed by a mathematical solver. To deal with incoming requests a periodic re-optimization (cf. [2]) is performed in the

second stage of TSF; i.e. the ABRE is repeated at given times (referred to as planning updates). Let t identify an individual planning update. Each planning update includes the following steps: (i) the planning relevant data (e.g. request pools, etc.) are updated; (ii) each agent computes the own bid generation; (iii) the mediator reallocates incoming requests by winner determination; (iv) the agents update their round trips by resolving the TPP; (v) the mediator calculates Shapley values.

Our TSF suggests that each agent calculates a bid for any available request-cluster (composition of incoming requests) in his bid generation (step (ii)). For the bid generation, the earnings regarding the incoming requests of any request-cluster are reduced by the increased transportation costs that each agent approximates by the cheapest insertion algorithm of [3]. It means that the round trips of all agents are extended by the incoming requests of a request-cluster (i.e. starting with the request with lowest costs). For the bid generation, the difference of costs between the original and the extended round trips are supposed as bids for all request-clusters.

The WDP (step (iii)) is formulated as a set partitioning problem (SPP; Eq. (10)–(14)). Let B store all request-clusters, while $b \in B$ identifies a specific request-cluster. The constant $d_{bi} = 1$ shows that incoming request i belongs to request-cluster b; otherwise $d_{bi} = 0$ applies. The bids g_{pb} result from the bid generations. To ensure an equality between transferred and received requests, we introduce a lower bound lb_p and an upper bound ub_p for each agent $p \in P$. The bounds limit the number of incoming requests that each agent is able to receive through the ABRE. The binary decision variable y_{pb} is equal to one if agent p wins request-cluster b and zero, otherwise. Objective (10) maximizes the sum of winning bids w by respecting that each incoming request i is reallocated once (Constraints (11)). While Constraints (12) observe that each agent wins only one request-cluster, Constraints (13) ensure that the sum of reallocated incoming requests lies between lb_p and ub_p for all agents p. Constraints (14) define the domains of the decision variables.

$$max \quad w = \sum_{p \in P} \sum_{b \in B} g_{pb} \cdot y_{pb}, \tag{10}$$

$$s.t. \quad \sum_{p \in P} \sum_{b \in B} y_{pb} \cdot d_{bi} = 1, \qquad \forall i \in N_c, \tag{11}$$

$$\sum_{b \in B} y_{pb} = 1, \qquad \forall p \in P, \tag{12}$$

$$lb_p \leq \sum_{i \in N_c} \sum_{b \in B} y_{pb} \cdot d_{bi} \leq ub_p, \qquad \forall p \in P, \tag{13}$$

$$y_{pb} \in \{0, 1\}, \qquad \forall p \in P, b \in B. \tag{14}$$

To improve the performance of our approach, we introduce a dynamic adjustment (DA) of the lower and upper bounds for the individual planning updates (Eqs. (15)–(16)). Thereby, the bounds are recalculated for all planning updates according to the number of received incoming requests of the previously executed planning update. It means that an agent that receives numerous incoming requests through the ABRE of a planning updated will be restricted by strict bounds for the forthcoming

planning update. On the other hand, an agent that receives less incoming requests will be favored by increasing his lower and upper bound. Therefore, for each agent the number of received incoming requests of an actual planning update is determined. Let u_p^t denote the number of received requests of agent p and planning update t. Separately, mb_p^t stores the requests that agent p offered for exchange of planning update t. The lower bounds lb_p^{t+1} of all agents $p \in P$ for the forthcoming planning update $t+1$ are calculated by Eq. (15). Thereby, the actual lower bound lb_p^t of each agent p is reduced by the difference of u_p^t and mb_p^t. Simultaneously, the upper bounds ub_p^{t+1} for all agents p are updated by Eq. (16).

$$lb_p^{t+1} = lb_p^t - (u_p^t - mb_p^t), \qquad \forall p \in P, t \in T \tag{15}$$

$$ub_p^{t+1} = ub_p^t - (u_p^t - mb_p^t), \qquad \forall p \in P, t \in T \tag{16}$$

4 Computational Study

The restricted WDP is analyzed on new DCTSP-instances regarding collaborative profits and an even allocation of requests. We consider 40 instances with different parameter settings that are organized in 4 test sets. For a detailed description of the instances, we refer to our homepage.[1] All instances provide that all agents receive and offer the same number of incoming requests per planning update. To simulate the dynamic execution of the round trips, we suppose that each agent is able to dispatch mb_p^t requests in the time between two planning updates. Our TSF with the described solution methodology was implemented in a C++-application on a Windows 7 PC (3.4 GHz, 16 GB RAM). The mathematical solver CPLEX 12.5.1 was used to solve the agents' specific TPPs respectively the SPPs. To reduce the computational effort, the computing time was limited to 600 s per optimization.

Table 1 presents the aggregated results per test set. The test sets are repeated with different values of the bounds (lb_p, ub_p), respectively with and without DA. The amount of the collaborative profits $z(P)$ and the number of achieved best solutions *best* are specified, while an even allocation of incoming requests is analyzed by the minimal (r_{min}) and the maximal (r_{max}) number of dispatched requests by an agent during the whole planning period. The range $(|r_{max} - r_{min}|)$ results from the difference of both values. Since the solution space is not restricted, values of $lb_p = 0$ and $ub_p = \infty$ averagely achieve superior collaborative profits for all test sets. However, an uneven allocation of incoming requests cannot be excluded; i.e. maximal ranges. Even values of the bounds $(lb_p = ub_p = mb_p^t)$ exclude shifts in the request-pools of the agents. Thereby, a decrease of the collaborative profits between 2.6 and 9.4% has to be accepted. A balance between the collaborative profits and an even allocation of requests can be achieved by using different values for lb_p and ub_p. Thereby, the application of DA increases the solution quality regarding both the amount of

[1] http://www.logistik.uni-bremen.de/english/instances/.

Table 1 Results of the computational study

lb_p	ub_p	DA	Profits		Requests		
			$z(P)$	Best	r_{min}	r_{max}	Range
Test set 1 ($\|p\| = 4, \|t\| = 10, m_p^t = 4$)							
0	∞	No	**5,377.5**	**3**	27.4	78.3	50.9
4	4	No	5,237.0	1	49.0	49.0	0.0
3	5	No	5,252.7	0	44.0	54.9	10.9
2	6	No	5,309.7	1	35.4	61.8	26.4
1	7	No	5,294.5	0	32.9	65.7	32.8
3	5	Yes	5,356.7	**3**	47.2	50.4	3.2
2	6	Yes	5,330.9	2	46.5	51.5	5.0
1	7	Yes	5,216.1	0	45.7	52.4	6.7
Test set 2 ($\|p\| = 5, \|t\| = 10, m_p^t = 3$)							
0	∞	No	**5,462.2**	**4**	15.5	67.7	52.2
3	3	No	5,129.1	0	39.0	39.0	0.0
2	4	No	5,279.1	0	33.0	45.9	12.9
1	5	No	5,321.2	0	24.7	54.0	29.3
0	6	No	5,368.7	1	16.9	58.2	41.3
2	4	Yes	5,220.8	1	36.2	40.6	4.4
1	5	Yes	5,270.2	1	35.6	42.0	6.4
0	6	Yes	5,345.7	3	34.0	43.0	9.0

(continued)

Table 1 (continued)

lb_p	ub_p	DA	Profits		Requests		
			$z(P)$	Best	r_{min}	r_{max}	Range
Test set 3 ($\|p\| = 3, \|t\| = 15, m_p^t = 5$)							
0	∞	No	**7,309.2**	**7**	30.0	137.2	107.2
5	5	No	6,619.9	0	84.0	84.0	0.0
4	6	No	6,722.6	0	78.4	89.3	10.1
3	7	No	6,625.9	0	71.2	97.9	26.7
2	8	No	6,769.3	0	59.2	109.9	50.7
4	6	Yes	6,909.9	1	82.9	84.9	2.0
3	7	Yes	6,769.9	1	82.0	86.2	4.2
2	8	Yes	6,892.0	1	78.9	87.7	8.8
Test set 4 ($\|p\| = 3, \|t\| = 10, m_p^t = 5$)							
0	∞	No	**4,951.1**	2	25.6	98.9	73.3
5	5	No	4,724.5	0	59.0	59.0	0.0
4	6	No	4,736.9	0	54.6	63.8	9.2
3	7	No	4,574.1	0	49.2	68.1	18.9
2	8	No	4,794.6	0	43.6	74.5	30.9
4	6	Yes	4,800.4	2	57.9	60.1	2.2
3	7	Yes	4,888.2	**4**	56.3	60.9	4.6
2	8	Yes	4,843.1	2	55.6	61.4	5.9

collaborative profits and mean ranges. Particularly, the parameter setting $lb_p = m_p^t - 1$ and $ub_p = m_p^t + 1$ with DA achieves an excellent balancing between both aims; i.e. decrease of collaborative profits between 0.4 and 5.5% against an unrestricted WDP and ranges between 2.0 and 4.4 requests.

The results of our computational study verify that lower and upper bounds for the WDP can enforce an even allocation of incoming request. Thereby, our approach is appropriate for dynamic ABREs, while the solution quality can be increased by the application of DA. The improved stability for horizontal coalitions may absorb the slightly lower collaborative profits against an unrestricted WDP. In this paper a first study on the restricted WDP in case of DCTSPs has been performed. To transfer our results to the daily transport business of SMCs, further computational experiments on more realistic scenarios have to be performed.

Acknowledgements The research was supported by the German Research Foundation (DFG) as part of the project "Kooperierende Rundreiseplanung bei rollierender Planung".

References

1. Feillet, D., Dejax, P., Gendreau, M.: Traveling salesman problems with profits. Transp. Sci. **39**(2), 188–205 (2005)
2. Pillac, V., Gendreau, M., Guret, C., Medaglia, A.L.: A review of dynamic vehicle routing problems. Eur. J. Oper. Res. **225**(1), 1–11 (2013)
3. Rosenkrantz, D.J., Stearns, R.E., Lewis, P.M.: An analysis of several heuristics for the traveling salesman problem. SIAM J. Comput. **6**(3), 563–581 (1977)
4. Schwind, M., Gujo, O., Vykoukal, J.: A combinatorial intra-enterprise exchange for logistics services. Inf. Syst. E-Bus. Manag. **7**(4), 447–471 (2009)
5. Shapley, L.: A value for n-person games. In: Kuhn, H., Tucker, A. (eds.) Contributions to the Theory of Games II 28, pp. 307–317. Princeton University Press, Princenton, NJ (1957)
6. Wang, X., Kopfer, H.: Rolling horizon planning for a dynamic collaborative routing problem with full-truckload pickup and delivery requests. Flex. Serv. Manuf. J. **27**(4), 1–25 (2015)

Comparing Two Optimization Approaches for Ship Weather Routing

Laura Walther, Srikanth Shetty, Anisa Rizvanolli and Carlos Jahn

Abstract Weather routing in maritime shipping is related to a shipping company's objective to achieving maximum efficiency, economy and cost competitiveness by optimizing each voyage of a ship. A voyage can be optimized regarding cost, time, safety or a combination of these factors, while considering forecasted meteorological and oceanographic information as well as constraints given by geographic conditions, ship characteristics, emission regulations, safety requirements or time restrictions. A wide variety of mathematical models of the ship weather routing problem as well as different approaches to solve it can be found in the literature and are applied by numerous software systems. This paper presents two approaches to solve the ship weather routing problem, a graph algorithm and an evolutionary approach. Both approaches aim to minimize fuel costs, allowing for route and speed optimization. They are compared based on numerical examples with real-world data.

1 Ship Weather Routing Problem

Voyage planning and optimization represents a widespread measure to improve cost and energy efficiency of maritime shipping. Ship weather routing generally aims to find an optimal route and speed profile for a ship's voyage based on the analysis of metocean weather forecasts. Meteorological institutes commonly use the mathematically concise data format GRIB (General Regularly-distributed Information in Binary form) to store weather data numerically predicted for each node of a grid. The ship weather routing problem is mathematically modeled in various ways [16].

L. Walther (✉) · S. Shetty · A. Rizvanolli · C. Jahn
Fraunhofer CML, Hamburg, Germany
e-mail: Laura.Walther@cml.fraunhofer.de

S. Shetty
e-mail: Srikanth.Shetty@cml.fraunhofer.de

A. Rizvanolli
e-mail: Anisa.Rizvanolli@cml.fraunhofer.de

C. Jahn
e-mail: Carlos.Jahn@cml.fraunhofer.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_45

Formulations not only range from one to multiple objective optimization problems, but also from constrained graph problems to nonlinear optimization problems. In order to solve the optimization problem, different approaches are applied by several commercial systems which now support voyage optimization on vessels, as well as by numerous academic software developments. These vary from calculus of variations [2], dynamic programming [1, 6, 12] or graph algorithms [4, 7, 14] to evolutionary approaches [5, 9, 13]. Superiority of an approach producing satisfactory results with adequate computational effort significantly depends on the degree to which the specific requirements regarding optimization objectives, variables, constraints and implementation are met [16]. For the ship weather routing problem described below, two popular approaches, a graph algorithm and an evolutionary method, are presented, compared and discussed.

Objective Function The objective is either minimum fuel costs, minimum voyage time, or maximum safety, or these objectives are combined giving rise to a multi-objective problem. As cost and energy efficiency are key aspects in maritime shipping, in this study the objective is minimum fuel costs C_{Fuel}.

Variables To allow route and speed optimization, the ship's heading α_G and speed over ground v_G are introduced as control variables. A certain speed requires variable engine power considering different environmental impacts. Speed and weather conditions are assumed to be constant between two waypoints of the ship's route.

Constraints Constraints on the variables are given by the ship itself, by time, safety and geographic restrictions. For simplicity reasons, safety constraints such as critical wave heights or periods are neglected. Geographic constraints primarily refer to land, but can also include traffic separation schemes, icebergs or mines. As a deep sea voyage is assessed, these constraints are not further elaborated. Time restrictions are most likely related to the estimated time of arrival (ETA). A certain arrival time $t_{Arrival}$ is assumed to be obligatory. Referring to constraints due to ship characteristics, the ship's design and propulsion system influence its behavior, speed profile and fuel consumption when facing environmental impacts such as waves or wind. Considered constraints include a maximum speed through water due to a maximum power of the ship's engine and a minimum speed to maintain course control.

2 Optimization Approaches

The ship weather routing problem as described above is a single-objective deterministic and constrained optimization problem. It is approached below using a graph algorithm and a genetic algorithm. Both approaches aim to minimize fuel costs, while varying the ship's heading and speed to allow route and speed optimization.

Graph Algorithm The described ship weather routing problem is discretized in time and space. An according graph is used, which is connected, directed and acyclic [15]. A common deterministic method for solving a discrete single-objective optimization

problem related to finding the optimal path in a graph is Dijkstra's algorithm [3], which is applied in ship weather routing [11, 14]. To reduce computational effort the A* algorithm is applied in this study [15]. An optimal path in this case is the path of minimum fuel costs, thus the arc weights are the fuel costs between the two respective nodes. It is aimed to minimize the total estimated costs $F(k)$, which is the sum of the exact fuel costs $G(k)$ according to Sect. 3 from the start to any node k and the heuristic estimated fuel costs $H(k)$ from k to the destination, which are derived equivalently to $G(k)$ but neglecting the predicted weather conditions. The selection criterion is expressed in Eq. (1) with B denoting a set of nodes not considered on the route from start to k [15].

$$F(k) = G(k) + H(k) \leq \min\{G(i) + H(i) \mid i \in B\} \tag{1}$$

Genetic Algorithm Evolutionary methods, mainly genetic algorithms (GA), are becoming increasingly popular as it is more often aimed at decision support by solving a multi-objective optimization problem [5, 13]. The objective is to find the route r^j of minimum fuel costs $C_{Fuel}(r^j)$ from the set of all feasible routes R. A route's fuel costs are the sum of the costs between two neighboring waypoints i and $i+1$ with $r^j = \{x_1^j, x_2^j, \ldots, x_n^j, y_1^j, y_2^j, \ldots, y_n^j, v_1^j, v_2^j, \ldots, v_n^j\}$ being a vector of decision variables describing the waypoints (x_i^j, y_i^j) and the speed profile (v_i^j). To apply the GA in this case, it is made use of the GA from the optimization toolbox of Matlab R2016a, which is integrated in the C++ framework. An initial population $r^{initial}$ is given for each voyage (see Sect. 4). Using the GA default selection, reproduction, crossover and mutation mechanisms further generations are created until a local optimal solution is provided [8].

3 Ship Hydrodynamics and Calculation of Fuel Costs

The optimization aims to minimize fuels costs. These can be derived based on time- and location-dependent meteorological and oceanographic impacts, especially ocean currents, wind and waves, as well as the ship's characteristics, mainly resistance and propulsion system. As the current is neglected in this study, the ship's speed v_S and heading α_S through water are equal to those over ground. The same applies to true wind speed u_T and direction α_T and those relative to the ground.

Ship Resistance The total resistance of a ship R_{total} is composed of its resistance in calm water R_{Calm} and an added resistance influenced by the ship's roughness and appendages as well as environmental impacts [10]. Here, the added resistances due to wind R_{Wind} and waves R_{Wave} are considered, as in Eq. (2). Wind speed u_T and direction α_T as well as wave period T_W, direction μ_0 and height H_S are given in weather forecasts, while ship speed v_S and heading α_S are variables.

$$R_{total} = R_{Calm}(v_S) + R_{Wind}(u_T, \alpha_T, v_S, \alpha_S) + R_{Wave}(T_S, \mu_0, H_S, v_S, \alpha_S) \tag{2}$$

Calm Water Resistance The calm water resistance R_T of a ship can be derived amongst others from model tests or empirical formulae. It can be expressed as a polynomial function of the ship's speed through water v_S, as in Eq. (3).

$$R_{Calm}(v_S) = a_4 v_S^4 - a_3 v_S^3 + a_2 v_S^2 - a_1 v_S + a_0 \tag{3}$$

Added Resistance due to Wind Due to the effect of the true wind speed u_T at an angle α_T, the ship's speed v_S and heading α_S, the ship experiences an apparent wind speed u_A. To estimate the wind resistance R_{Wind} the simplified approach in Eq. (4) is used that depends on the apparent wind along the ship's center line $u_{A,S} = v_S + u_T \cdot \cos(\alpha_T - \alpha_S)$, the ship's frontal projected area above sea level A_F, the density of air ρ_{Air} and a coefficient c_A, which is 0.8–1.0 for cargo ships [10]. Accordingly, head wind causes an additional resistance, while tailwind reduces the ship's resistance.

$$R_{Wind}(u_T, \alpha_T, v_S, \alpha_S) = \begin{cases} 0.5 \cdot \rho_{Air} \cdot c_A \cdot A_F \cdot u_{A,S}^2 & , \quad u_{A,S} \geq 0 \\ - \; 0.5 \cdot \rho_{Air} \cdot c_A \cdot A_F \cdot u_{A,S}^2 & , \quad u_{A,S} < 0 \end{cases} \tag{4}$$

Added Resistance due to Waves The added resistance R_{Wave} can be derived from hydrodynamic calculations. It depends on wave period T_w, encounter angle between ship and wave μ_e, wave height H_S and ship speed v_S. The encounter angle μ_e is the angle between main wave direction μ_0 and ship's heading α_S. Here, the added resistance $R_{Wave,H}$ standardized with the square of the wave height H_S is given in a matrix used to interpolate the added resistance due to waves $R_{Wave}(T_S, \mu_0, H_S, v_S, \alpha_S)$.

Engine Power and Fuel Consumption Accounting for the ship's propulsion system, the ship's resistance results in a required engine power, the fuel consumption and finally the costs of the route. Total resistance R_{total}, ship speed v_S and propulsion efficiency η_D compose the delivered shaft power with a corresponding specific fuel consumption $b_{e,Fuel}$. Combined with voyage time t and price per ton of heavy fuel oil P_{Fuel} it leads to the fuel costs C_{Fuel} as per Eq. (5), which are the time- and space-dependent arc weights of the graph. Losses in shaft or bearings are neglected.

$$C_{Fuel} = \frac{R_{total} \cdot v_S}{\eta_D} \cdot b_{e,Fuel} \cdot t \cdot P_{Fuel} \tag{5}$$

4 Comparison of Results, Discussion and Conclusions

The two approaches are compared based on transatlantic voyages of a bulk carrier transporting coal from Venezuela to Europe using weather forecasts from 2013-12-16. The ship has a length between perpendiculars of 220 m, a breadth of 32.24 m, a draught of 14.5 m, a displacement of 90,617 t and an engine power available for propulsion of 17,240 kW. The weather data covers the Atlantic ocean with a latitudinal and longitudinal resolution of 0.25°, a temporal resolution of 3 h and a fore-

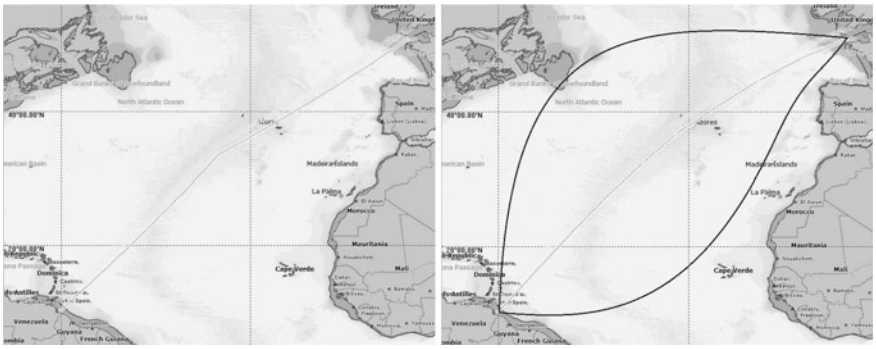


Fig. 1 Transatlantic voyage of bulk carrier from Venezuela to the English Channel within 12 days. Left side shows result from graph algorithm and right from genetic algorithm including boundaries

cast range of 7.5 days. The described objective, constraints, variables, implementation in C++ and system settings are considered to allow direct comparison of both approaches regarding computation time and quality of results. To allow on-board voyage optimization an ordinary personal computer is used.

Comparison A scenario with a minimum speed of 5 kn, a maximum speed of 15 kn and a voyage duration of 12 days is solved using the A* algorithm. For the duration outside the forecast range, the shortest distance is assumed. The result shown in Fig. 1 is achieved in less than one hour. This scenario is used as baseline for comparison. The time consuming part of the computation is the calculation of fuel costs, hence the arc weights, due to the consideration of 130 neighbors described by latitude, longitude and time. Assuming a variable arrival time and a constant speed which eliminates the time discretization, the computation time is less than one minute. Halving the geographic resolution returns a result in 7% of the baseline computation time, while halving the geographic resolution and simultaneously doubling the temporal resolution requires approximately 50% of the baseline computation time. Distance and fuel costs differ by less than 5% compared to the baseline. As to the genetic algorithm, an initial population is given by the Great Circle Route (GCR) and an average speed of 13 kn. An upper (UB) and lower boundary (LB) are displayed in Fig. 1. ETA, minimum and maximum speed are the same as above. A population size of 20 and 30 variables describing route and speed profile results in the route shown in Fig. 1, but takes 37% more time than the baseline, thus more than one hour. Distance and fuel costs are almost equal to the baseline. Decreasing the number of variables to 18 reduces time by 30% compared to the baseline without impairing distance and costs. When setting the LB to the initial population the result is not acceptable as it does not resemble the minimum found with the A* algorithm or with the GCR as the initial population. The results are not improved when using LB, 18 variables and a population size of 50. Only increasing the population size to 100 results in a good output in this case, but this also leads to a seven times higher computation time.

Further tests regarding mutation rate, crossover mechanisms or other options would be interesting, but are not addressed in this study.

Discussion and Conclusions The graph algorithm is mainly influenced by the discretization in space and time. As expected, the results of the genetic algorithm strongly depend on initial population, population size and number of variables. A suitable initial population with a small number of variables provides sound results in adequate computation time, even at a rather small population size. However, when it comes to initial populations not close to the optimum, population size needs to be increased significantly implying a major rise in computation time. First, the optimum cannot always be predicted to set the initial population accordingly, but a variation of the initial population may contribute to decision support. Second, bearing in mind that updated weather forecasts can be provided e.g. every 6 h, computation time needs to be as short as possible. Consequently, due to more reliable results that do not depend as strongly on the input data, the graph algorithms is considered to be advantageous for the described problem and application.

References

1. Avgouleas, K.: Optimal Ship Routing. MIT, Cambridge (2008)
2. Bijlsma, S.K.: On Minimal-Time Ship Routing. University of Technology Delft, Delft (1975)
3. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
4. Hagiwara, H.: Weather Routing of (Sail-Assisted) Motor Vessels. University of Technology Delft, Delft (1989)
5. Hinnenthal, J.: Robust Pareto-Optimum Routing of Ships utilizing Deterministic and Ensemble Weather Forecasts. Technische Universität Berlin, Berlin (2008)
6. Klompstra, M.B., Olsder, G.J., van Brunschot, P.K.G.M.: The isopone method in optimal control. Dyn. Control. **2**(3), 281–301 (1992)
7. Lin, Y.-H., Fang, M.-C., Yeung, R.W.: The optimization of ship weather-routing algorithm based on the composite influence of multi-dynamic elements. Appl. Ocean Res. **43**, 184–194 (2013)
8. MathWorks: MATLAB and Simulink—Genetic Algorithm Options. https://de.mathworks.com/help/gads/genetic-algorithm-options.html (2016). Accessed 29 Nov 2016
9. Marie, S., Courteille, E.: Multi-objective optimization of motor vessel route. Int. J. Mar. Navig. Saf. Sea Transp. **3**(2), 133–141 (2009)
10. Schneekluth, H., Bertram, V.: Ship Design for Efficiency and Economy. Butterworth-Heinemann, Oxford (1998)
11. Sen, D., Padhy, C.P.: An approach for development of a ship routing algorithm for application in the North Indian Ocean region. Appl. Ocean Res. **50**, 173–191 (2015)
12. Shao, W., Zhou, P., Thong, S.K.: Development of a novel forward dynamic programming method for weather routing. J. Mar. Sci. Technol. **17**(2), 239–251 (2012)
13. Szlapczynska, J.: Multi-objective weather routing with customised criteria and constraints. J. Nav. **68**(02), 338–354 (2015)
14. Takashima, K., Mezaoui, B., Shoji, R.: On the fuel saving operation for coastal merchant ships using weather routing. Int. J. Mar. Navig. Saf. Sea Transp. **3**(4), 401–406 (2009)
15. Turau, V.: Algorithmische Graphentheorie. Oldenbourg Verlag, München (2009)
16. Walther, L., Rizvanolli, A., Wendebourg, M., Jahn, C.: Modeling and optimization algorithms in ship weather routing. Int. J. e-Navig. Marit. Econ. **4**, 031–045 (2016)

Optimal Dynamic Assignment of Internal Vehicle Fleet at a Maritime Rail Terminal with Uncertain Processing Times

Ying Xie and Dong-Ping Song

Abstract This study aims to improve the efficiency of container loading process at a seaport by optimizing the dynamic assignment of internal vehicle fleet in the process of moving containers from storage yards at maritime terminals to the train at the rail terminal. We formulate the problem into a stochastic dynamic programming model taking into account uncertain processing times. Numerical experiments based on a case study are performed to illustrate the effectiveness and the sensitivity of the model.

1 Introduction

The growing traffic volume puts a huge pressure on container port as an interface between seaborne transport and hinterland transport. Rail transport is regarded as an effective way to tackle the above challenges due to its high capability and low emission. Therefore, improving the efficiency of rail terminal operations at seaports is essential to ensure the sustainability of global container transport chains. This study aims to improve the efficiency of container loading process at a seaport by optimizing the dynamic assignment of internal vehicle fleet in the process of moving containers from storage yards at maritime terminals to the train at the rail terminal.

A number of survey papers have reviewed operations management at container ports and terminals, e.g. Stahlbock and Voss [6]; Carlo et al. [4]. However, the operations management issues directly associated with rail terminals at seaports

Y. Xie (✉)
Lord Ashcroft International Business School, Anglia Ruskin University,
Bishop Hall Lane, Chelmsford CM1 1SQ, UK
e-mail: ying.xie@anglia.ac.uk

D.-P. Song
School of Management, University of Liverpool, Chatham Street,
Liverpool L69 7ZH, UK
e-mail: Dongping.song@liverpool.ac.uk

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_46

have been understudied. A few papers focused on the container loading problem, which aims to assign containers to the wagon slots of the train by minimizing the unproductive operations at the rail terminal and/or in the storage areas [1, 2]. Caballini et al. [3] developed a mixed integer linear mathematical programming model to optimize the timings of the trains and the use of the handling resources devoted to rail port operations. The authors further extended the deterministic model to deal with unexpected situations or uncertainty by adopting an event-triggered receding-horizon planning approach. Their model does not consider the regular uncertainty in the container processing times.

There is a high level of uncertainty/variability in the process of moving containers from storage yards to the rail terminal. A need has emerged for tools that have the capability of appropriately determining the dynamic internal vehicle assignment in order to load containers onto the train within the time window. In this paper, we focus on the container loading process at a seaport from storage yards to trains. We will formulate the problem into a stochastic dynamic programming model, with the aim to minimize the total logistics costs associated with moving containers from storage yards to the train plus the penalty cost of underutilizing the train capacity.

2 Model and Solution

The process of transporting containers from storage yards to the train includes the following main activities (see Fig. 1): Internal Moving Vehicle (IMV) receives a message to collect a container; the container is landed on the IMV; IMV transports the container to the rail terminal (either to the Rail Terminal (RT) buffer area before the working time window, which is called pre-staging, or to the Rail Mounted Gantry crane (RMG) directly during working time window); the pre-staged containers are moved from the RT buffers to the RMG; RMG loads the container to a wagon slot on the train.

Consider the loading process of a single train under periodic-review scheme with a working time window $(0, T)$. The decision variables include: q: the number of containers to be pre-staged from storage yards to the rail terminal (RT) buffer before the working time window; $u^V(t)$: the planned flow rate (i.e., the number of assigned

Fig. 1 The process of transporting containers from yards to train

Table 1 Notation of static parameters

T:	The planning horizon, assuming the working time window is (0, T)
Q^B:	The capacity of the RT buffer space
Q^C:	The maximum handling capacity of the RMG within one period
Q^T:	The capacity of the train
c^P:	The unit cost of pre-staging containers (including transport and storage)
c^V:	The unit cost of vehicle deployed to transport a container from yard to RMG
c^B:	The unit cost of vehicle deployed to transport a container from RT buffer to RMG
c^C:	The unit cost of the RMG loading a container to train
c^S:	The storage cost at RT buffer per container per period
c^U:	The unit penalty cost of underutilizing the train capacity

Table 2 Notation of dynamic parameters and variables

$U^V(t)$:	The maximum number of assigned IMV from yard to RMG in period t
$U^B(t)$:	The maximum number of assigned IMV from RT buffer to RMG in period t
$\xi^V(t)$:	The random flow rate from yard to RMG in period t
$\xi^B(t)$:	The random flow rate from RT buffer to the RMG in period t
$x^B(t)$:	The number of containers in the RT buffer at the end of time period t
$x^T(t)$:	The number of containers on the train at the end of time period t

IMV) to move containers from yards to the RMG over the working time window; and $u^B(t)$: the planned flow rate (i.e., the assigned number of IMV) to move containers from RT buffer to the RMG over the working time window. We assume that one IMV carries one container. Other parameters are introduced and shown in Tables 1 and 2. The objective is to minimize the total cost incurred during pre-staging containers, transporting containers from yards to RMG, transporting containers from RM buffer to RMG, RMG crane handling containers, container storage at buffers, and penalty for underutilizing the train capacity.

2.1 Model

The discrete-time dynamics of the transportation system can be described by

$$x^B(t) = x^B(t-1) - \xi^B(t), \text{for } t = 1, 2, \ \ldots, \ T; \tag{1}$$

$$x^T(t) = x^T(t-1) + \xi^V(t) + \xi^B(t), \text{for } t = 1, 2, \ \ldots, T. \tag{2}$$

$$x^B(0) = q; x^T(0) = 0; 0 \le q \le Q^B; \tag{3}$$

$$0 \le \xi^V(t) \le u^V(t); 0 \le \xi^B(t) \le u^B(t); \tag{4}$$

$$0 \le u^V(t) \le U^V(t); \quad 0 \le u^B(t) \le \min\left(x^B(t-1), U^B(t)\right); \tag{5}$$

$$u^V(t) + u^B(t) \le \min\left(Q^T - x^T(t-1), Q^C\right); \tag{6}$$

The initial number of containers in the RT buffer is q, and the initial number of containers on the train is 0. It should be noted that due to the uncertainty in container processing time, the actual number of containers that reach the RMG in one period, represented by $\xi(t)$, is often lower than the planned flow rate $u(t)$. Thus we have constraints in (4). The planned flow rate $u(t)$ is also constrained by the maximum number of IMVs available, by the capacity of the train and by the capacity of the RMG, as shown in (5, 6).

The objective function is given by:

$$\begin{aligned} J_0(q,0,0) = \mathrm{E}[q \cdot c^P + \textstyle\sum_{t=0}^{T} c^S x^B(t) + \sum_{t=1}^{T} \left(c^V u^V(t) + c^B u^B(t) + c^C\left({}^B(t) + {}^V(t)\right)\right) \\ + c^U \cdot (Q^T - \xi^T(T))] \end{aligned} \tag{7}$$

On the right-hand-side of the above equation, the first term is the pre-staging cost; the second term is the storage costs at RT buffer; the third term represents the container movement costs from yard to RMG, from RT buffer to RMG, from RMG to train; the fourth term represents the penalty cost for underutilizing the train capacity. Following the stochastic dynamic programming theory [5], the backwards optimality equation is given by (for $t = 0, 1, \ldots, T$):

$$\begin{aligned} J_t\left(x^B(t), x^T(t)\right) = \min\{ & q \cdot c^P \cdot \mathrm{I}\{t=0\} + c^S x^B(t) + c^V u^V(t+1) \\ & + c^B u^B(t+1) c^U \cdot \left(Q^T - x^T(t)\right) \cdot \mathrm{I}\{t=\mathrm{T}\} \\ & + \mathrm{E}[c^C(\xi^B(t+1) + \xi^V(t+1)) + J_{t+1}\left(x^B(t+1), x^T(t+1)\right)]\} \end{aligned} \tag{8}$$

where $J_{T+1}(x^B(T+1), x^T(T+1)) = 0$, and $\mathrm{I}\{\text{condition}\}$ is an indicator function. It takes 1 if the condition in $\{\}$ is true, 0 otherwise.

2.2 Solution

The stochastic dynamic programming problem in (1)–(8) can be solved using the backwards value iteration algorithm (c.f. [5]).

Step 1: Let $J_{T+1}(x^B, x^T) = 0$ for any (x^B, x^T). Let $t = T$.

Step 2: Use (8) to calculate the optimal value function $J_t(x^B(t), x^T(t))$ subject to (1)–(6), and the optimal control $u_t^V(x^B(t), x^T(t))$ and $u_t^B(x^B(t), x^T(t))$.

Step 3: Let $t = t - 1$. If $t \geq 0$, go to Step 2.
Step 4: Identify the optimal q^*. Return the optimal cost $J_0(q^*, 0)$; the optimal decision variables $q^*, u_t^V(x^B(t), x^T(t))$, $u_t^B(x^B(t), x^T(t))$.

3 Numerical Examples

In this section, we first provide an empirical case to demonstrate the container loading process at a seaport rail terminal and calibrate the input data. Secondly, we perform a range of experiments to illustrate the application of the proposed models.

Figure 2 shows the empirical data of container loading rates at a real rail terminal within a day (from a real case study in the UK). In total six trains are handled within a day, and each time period is 30 min. The number of containers handled per period ranges from 0 to 15. The working time window for each train ranges from 4 periods (i.e. 2 h) to 8 periods (i.e. 4 h). We calibrate the input data of the reference scenario as follows: the time period is 30 min; $Q^T = 40$; $Q^B = 30$; $Q^C = 15$; $U^V(t) = 15$; $U^B(t) = 15$. We assume $x^B(t) \equiv u^B(t)$ and $\xi x^V(t) = u^V(t) \cdot z$, where z follows a uniform distribution. Here we want to focus on the uncertainty in the process from yards to rail terminal by assuming deterministic operations from RT buffer to RMG. Moreover, let $c^P = 4$; $c^V = 5$; $c^B = 2$; $c^C = 1$; $c^S = 1$; $c^U = 100$. It should be noted that the above cost coefficients are hypothetical and only the relative values of these cost elements are meaningful.

Now we apply the model to optimize the pre-staging decision and the dynamic IMV assignment. As the length of working time window is an important factor, we experiment with three levels of working window, i.e. $T = 4$, 6, 8, which correspond to 2, 3, and 4 h working windows respectively. The results are given in Table 3.

From Table 3, it can be seen that: (i) in the deterministic situation, we have $q^* = 0$, which means zero pre-staging is optimal. This is intuitively true due to the facts: (a) pre-staging plus moving containers from RT buffer to RMG costs more than directly moving containers from yard to RMG; (b) the working time window is

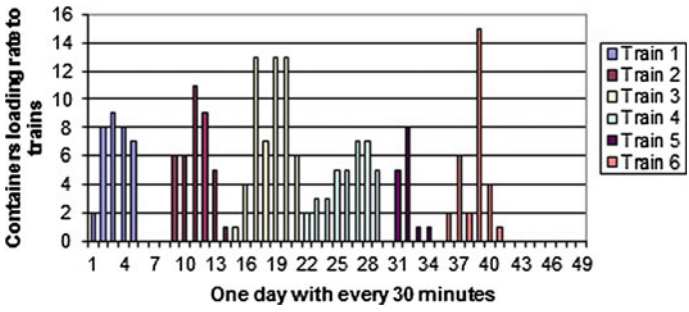


Fig. 2 Empirical data of container loading rate at a rail terminal within a day

Table 3 Optimizing IMV assignment and pre-staging

	T = 4		T = 6		T = 8	
z	q^*	$J_0(q^*, 0)$	q^*	$J_0(q^*, 0)$	q^*	$J_0(q^*, 0)$
U(1, 1)	0	240.00	0	240.00	0	240.00
U(0.8, 1)	3	283.05	1	270.80	0	267.95
U(0.6, 1)	12	334.78	1	300.70	0	297.44
U(0.4, 1)	30	363.63	17	340.45	16	334.25
U(0.2, 1)	30	394.35	30	372.42	30	361.29

sufficiently large to move containers directly from yards to RMG to fully load the train; (ii) $J_0(q^*, 0)$ is increasing in the degree of uncertainty; and q^* is increasing in the degree of uncertainty; (iii) by comparing the results with that of zero pre-staging cases (not included in this paper due to page limit), the cost saving of the best pre-staging decision from zero pre-staging is increasing as the degree of uncertainty increases. This indicates the importance of determine appropriate pre-staging. (iv) At the same degree of uncertainty, $J_0(q^*, 0)$ is decreasing as the time window increases; and q^* is decreasing as the time window increases. When the time window is adequately large, zero pre-staging tends to be optimal.

4 Conclusions

This study considers the optimal assignment of IMV fleet and container pre-staging at a seaport rail terminal in the presence of uncertainty. The mathematical model developed using stochastic dynamic programming can plan the container flow at aggregate level, without the need to address the detailed discrete events, therefore can avoid the NP hard combinational optimization problem. Another innovation of the developed model is the ability of yielding optimal plans under dynamic mode and accommodating stochastic factors. However, when the dimension of state and decision variables increases, the computation complexity of the model also increases significantly. Numerical examples based on a real case are provided to illustrate the effectiveness of the model. Further research includes combining both discharge and load trains into a single optimization model.

References

1. Ambrosino, D., Siri, S.: Comparison of solution approaches for the train load planning problem in seaport terminals. Transp. Res. Part E **79**, 65–82 (2015)
2. Anghinolfi, D., Paolucci, M.: A general purpose lagrangian heuristic applied to the train loading problem. procedia—social behaviour. Science **108**, 37–46 (2014)

3. Caballini, C., Pasquale, C., Sacone, S., Siri, S.: A receding-horizon planning approach for rail operations in seaport container terminals. IEEE Trans. Intell. Transp. Syst. **15**(1), 365–375 (2014)
4. Carlo, H.J., Vis, I.F.A., Roodbergen, K.J.: Transport operations in container terminals: literature overview, trends, research directions and classification scheme. Eur. J. Oper. Res. **236**, 1–13 (2014)
5. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (2005)
6. Stahlbock, R., Voß, S.: Operation research at container terminals—a literature update. OR Spectrum **30**(1), 1–52 (2008)

The Capacitated Vehicle Routing Problem with Three-Dimensional Loading Constraints and Split Delivery—A Case Study

Junmin Yi and Andreas Bortfeldt

Abstract The capacitated vehicle routing problem with three-dimensional loading constraints (3L-CVRP) combines vehicle routing and three-dimensional loading with additional packing constraints concerning, for example, the stability of packed goods. We consider a logistics company that repeatedly has to pick up goods at different sites. Often, the load of one site exceeds the volume capacity of a vehicle. Therefore, we focus on the 3L-CVRP with split delivery and propose a hybrid algorithm for this problem. It consists of a tabu search procedure for routing and some packing heuristics with different tasks. One packing heuristic generates packing plans for shuttle tours involving special sites with large-volume sets of goods. Another heuristic cares for packing plans for tours with numerous sites. The hybrid algorithm is tested with a set of instances which differs from often used 3L-CVRP test instances and comes from real industrial data, with up to 46 sites and 1549 boxes to be transported. The algorithm yields good results within short computing times of less than 1 min.

1 Introduction

The capacitated vehicle routing problem with three-dimensional (3D) loading constraints (3L-CVRP) generalizes the vehicle routing problem and the container loading problem which are traditionally separately handled combinatorial optimization problems. Real-world settings can be modelled in greater detail by packing constraints which ensure the integrity of sensitive items, stability of packing arrangements and efficient unloading of delivered boxes.

Since the 3L-CVRP was introduced in [1], many effective algorithms proposed in literature are mostly hybrid metaheuristics. A nested tabu search algorithm is

J. Yi (✉)
Xiamen University of Technology, Xiamen, China
e-mail: yijunmin@xmut.edu.cn

A. Bortfeldt
Otto-von-Guericke University, Magdeburg, Germany
e-mail: andreas.bortfeldt@ovgu.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_47

developed by [1] and an ant colony algorithm is designed in [2]. Further effective hybrid algorithms are proposed in [3–6]. The literature on VRP with 3D loading constraints is surveyed in [7, 8].

In the 3L-CVRP, it is required that each customer is visited just once. However, in practice it is possible that a customer has a demand that does not fit into a single vehicle for reasons of weight or volume. In this case, the demand has to be split and be delivered by two or more vehicles. In the research dedicated to the classical VRP a large body of literature deals with VRP with split delivery (see [9]). But to our best knowledge, only the papers [10, 11] considered the possibility of splitting the customers' demands in a routing-packing problem context. However, these papers did not handle the situation where customer demands are larger than vehicle capacity. To fill this gap, our study addresses the 3L-CVRP with split delivery (3L-SDCVRP) in a milk-run operation of a Shanghai automotive logistics company.

The rest of the paper is organized as follows: the 3L-SDCVRP is formulated and related real-world instances are described in Sect. 2. Our solving approach to the 3L-SDCVRP is outlined in Sect. 3. Results are provided and discussed in Sect. 4. Conclusions are drawn in Sect. 5.

2 Problem Formulation and Shanghai Dataset

Our problem comes from the milk-run operations in and around Shanghai area that are carried out by a Shanghai automotive logistics company, which serves many car makers in metropolitan Shanghai and whole China. It can be formulated similarly to the 3L-CVRP (see [1, 10]).

Let be given a complete network with n nodes, one depot and symmetric distances. There is a fleet of homogeneous vehicles that are rear loaded and have identical 3D rectangular loading spaces. Each node has a pickup demand given by a set of 3D rectangular items. Our task is to determine a set of routes, starting and ending at the depot, and a packing plan for each route. The packing plan should stow all boxes, which are to be picked up at the nodes of the related route, in a feasible way (no overlapping items, each item must lie completely in the loading space, orthogonal packing). The pickup demands of all nodes have to be satisfied and the routes should be chosen so that the total travel distance is minimized and, as second objective criterion, the number of routes (or used vehicles) is as small as possible.

Moreover, some packing constraints have to be observed: (C1) *Loading sequence constraints*. Loading the items of a node must be possible by pure movements of these items in length direction of the vehicle. (C2) *Orientation constraints*. The spatial orientation of all items is fixed with regard to height while horizontal 90° rotations are permitted. (C3) *Support constraints*. A certain percentage a of the base area of all items must be supported by other items. We chose $a = 75\%$ in the experiments. (C4) *Fragility constraints*. Here, if a box type has three dimensions less or equal 100 cm and there is only one item of this type, the item may be classified as fragile. Fragile items can only bear other fragile items. A weight constraint is ignored here since all packed goods are of low density.

Table 1 Summary of the Shanghai dataset

Instance	Nodes (n)	Box types	Items (m)	Vehicle type	Minimum no. of vehicles (v_{LB})	No. of big nodes
Sha01	5	26	261	M	2	0
Sha02	8	50	167	S	6	2
Sha03	10	17	73	S	3	0
Sha04	12	33	204	B	3	1
Sha05	12	59	228	M	4	1
Sha06	15	56	228	M	4	1
Sha07	16	79	439	B	7	2
Sha08	18	51	303	M	6	1
Sha09	27	98	734	C	8	1
Sha10	31	134	590	B	9	1
Sha11	46	185	1549	C	16	4

The eleven problem instances are generated from the automotive logistics company, thus they are called here Shanghai dataset. Most instances include some nodes, called *big nodes*, whose demand exceeds the volume capacity of a vehicle so that (at least) for these nodes two or more routes are indispensable. Our instances have numbers of nodes (n) ranging from 5 to 46 and the numbers of items to be loaded (m) range from 73 to 1549, details are summarized in Table 1.

Although there are four vehicle types, only one type is chosen per instance. The lower bound v_{LB} for the number of vehicles (routes) is calculated as rounded quotient of the total items volume and the vehicles volume capacity.

Compared to the 3L-CVRP benchmark instances by [1, 4], our dataset has some important application-oriented attributes. The numbers of box types are quite large. The cargo of a node is often composed by large groups of items of same dimensions. The distance matrix is gained from the real-time road travel distances by Baidu e-map.

From the occurrence of big nodes in the Shanghai problems we can conclude that these problems are instances of 3L-SDCVRP. We assume in this paper that splits are only allowed when necessary, i.e. when the boxes of a node cannot be packed in one loading space. Note that we use the term 3L-CVRP with *split delivery* (instead of *split pickup*) since from a structural perspective there is no difference between these problem variants.

One could raise the question whether a big node cannot be replaced by two or more artificial nodes that have the same coordinates as the big node and the same total demand. In this case and if only inevitable splits are allowed, one could try to reduce the 3L-SDCVRP to the 3L-CVRP.

However, this procedure would require a split of the demand of each big node within the problem formulation. These anticipated splits would often be worse (in terms of solution quality) than splits generated by means of a packing algorithm. Hence, the 3L-SDCVRP seems to be an independent problem even if only inevitable splits are permitted.

3 Solving Approach for 3L-CVRP with Split Pickup

Our approach consists of two main steps and is based on two earlier published papers. In the first step, routes with only one or two nodes and related packing plans are constructed. This step is intended mainly for those nodes whose load almost reaches or exceeds the volume capacity of one vehicle. In the second step, the residual problem is solved by constructing routes with multiple nodes and related packing plans. The steps are described below with some details.

First main step: Packing plans for each node are generated by a genetic algorithm (GA) for the container loading problem that is proposed in [12]. Each packing plan consists of vertical layers that follow each other in length direction. The crossover operator generates an offspring by combining high quality layers from both parents and adding some newly constructed layers.

For each node the GA constructs at least one packing plan (one filled loading space) and two or more if necessary. Then pairs of packing plans of two nearby nodes will be merged to save some loading space. Finally, all packing plans are accepted that satisfy one of the following criteria: (i) the filling rate of the loading space reaches a given limit (e.g. 60%); (ii) the packing plan belongs to a series of at least two packing plans of the same node and does not have the worst filling rate of that series. An accepted packing plan is completed by a route (with one or two nodes) and the packed items and, if necessary, their nodes are removed from the problem instance.

Second main step: The remaining 3L-CVRP instance is solved by means of the hybrid algorithm developed in [3]. A tabu search algorithm serves for routing and performs swap as well as shift moves that include either one or two routes of a given solution. A tree search algorithm is responsible for packing checks. A packing plan for a route is built box by box in a backtracking manner and at each stage a small number of possible placements is examined. Much computational effort is saved by means of special coupling mechanisms between routing and packing, e.g., a cache which includes already tested routes.

However, the original tree search algorithm is only able to cope with small numbers of items and has been modified. Now, vertical layers, which fill the length or width of the loading space, can be integrated in packing plans yielded by the tree search algorithm; these layers are also produced by the above GA in step 1. In the end the best solutions of both steps are assembled.

4 Results and Discussion

Our hybrid algorithm has been implemented in C++ and tested on the above introduced dataset on a PC with an Intel processor (3.30 GHz). Detailed results are shown in Table 2; z stands for the total travel distance.

The reached mean filling rate per instance is given by the quotient (in %) of the total item volume and the total volume of the used loading spaces and is mainly responsible for the number of routes v. The mean filling rates are satisfactory with respect to the required constraints. Similar filling rates were achieved in, e.g. [1, 2]. Note that primarily the loading sequence constraint (C1) makes it difficult to reach larger filling rates in the 3L-(SD)CVRP (see [10], p. 1147). In our total 200 nodes of 11 instances, there are 14 big nodes in 9 instances. The number of nodes with split pickup exceeds the number of big nodes (see Table 1) only for three instances and by at most two nodes. This meets the practical requirement of "less splits, less management cost on sorting and counting".

To what extent nodes that are not "big" are also split, depends on the quality of the used packing algorithm. Thus, the small number of four additional splits in our results also indicates a good solution quality. By the way, the occurrence of additional splits shows again that the 3L-SDCVRP cannot be reduced to the 3L-CVRP even if only necessary splits are allowed.

All in all, we have reached good quality results and our solutions were provided in short running times of less than 1 min while in [10] (p. 1146) running times of nearly 3 h are reported.

Table 2 Summary of results

Instance	z	v	v_{LB}	Mean filling rate (%)	Number of nodes split	Running time (s)
Sha01	582.2	3	2	55.1	0	2
Sha02	2907.0	10	6	52.9	3	13
Sha03	369.2	4	3	53.6	1	42
Sha04	372.0	4	3	61.2	1	13
Sha05	1493.9	6	4	57.6	1	10
Sha06	620.0	7	4	55.7	1	19
Sha07	1701.4	11	7	60.9	2	28
Sha08	387.7	9	6	57.3	1	11
Sha09	1063.8	15	8	48.0	1	17
Sha10	1946.2	15	9	57.6	1	53
Sha11	581.8	29	16	53.0	6	33

5 Conclusions

We have considered the Shanghai dataset, a set of instances of the CVRP with three-dimensional loading constraints and split delivery that comes from the Shanghai automotive industry. We solved the problem under the assumption that only inevitable splits are allowed and showed that the 3L-SDCVRP under this assumption cannot be reduced to the 3L-CVRP and represents an independent problem. Our proposed hybrid algorithm effectively solves the Shanghai dataset in short running times.

Acknowledgements The authors would like to thank the China Society of Logistics and Anji Logistics for providing us the real-world data and for bringing to our attention this interesting problem. Also the supports from NSFC research grant 71371162 and Fujian Provincial science grant 2014J01271 are acknowledged.

References

1. Gendreau, M., Iori, M., Laporte, G., Martello, S.: A tabu search algorithm for a routing and container loading problem. Transp. Sci. **40**(3), 342–350 (2006)
2. Fuellerer, G., Doerner, K.F., Hartl, R.F., Iori, M.: Metaheuristics for vehicle routing problems with three-dimensional loading constraints. Eur. J. Oper. Res. **201**, 751–759 (2010)
3. Bortfeldt, A.: A hybrid algorithm for the capacitated vehicle routing problem with three-dimensional loading constraints. Comput. Oper. Res. **39**(9), 2248–2257 (2012)
4. Tarantilis, C.D., Zachariadis, E.E., Kiranoudis, C.T.: A hybrid metaheuristic algorithm for the integrated vehicle routing and three-dimensional container- loading problem. IEEE Transp. Intell. Transp. Syst. **10**(2), 255–271 (2009)
5. Wei, L., Zhang, Z., Lim, A.: An adaptive variable neighborhood search for a heterogeneous fleet vehicle routing problem with three-dimensional loading constraints. IEEE Comput. Intell. Mag. **9**, 18–30 (2014)
6. Zhang, Z., Wei, L., Lim, A.: An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints. Transp. Res. Part B **82**, 20–35 (2015)
7. Iori, M., Martello, S.: Routing problems with loading constraints. Top **18**, 4–27 (2010)
8. Pollaris, H., Braekers, K., Caris, A., Janssens, G.K., Limbourg, S.: Vehicle routing problems with loading constraints: state-of-the-art and future directions. OR Spectr. **37**(2), 297–330 (2015)
9. Toth, P., Vigo, D.: Vehicle Routing: Problems, Methods, and Applications, 2nd edn. Society for Industrial and Applied Mathematics, Philadelphia, PA (2014)
10. Ceschia, S., Schaerf, A., Stüzle, T.: Local search techniques for a routing-packing problem. Comput. Ind. Eng. **66**(4), 1138–1149 (2013)
11. Moura, A., Oliveira, J.F.: An integrated approach to the vehicle routing and container loading problems. OR Spectr. **31**(4), 775–800 (2009)
12. Bortfeldt, A., Gehring, H.: A hybrid genetic algorithm for the container loading problem. Eur. J. Oper. Res. **131**, 143–161 (2001)

A Model to Locate and Supply Bio-refineries in Large-Scale Multi-biomass Supply Chains

Nasim Zandi Atashbar, Nacima Labadie and Christian Prins

Abstract Biofuels derived from biomass can play a crucial role as one of the main sources of renewable energies. As logistics may represent up to 50% of biomass cost, it is necessary to design efficient biomass supply chains to provide bio-refineries with adequate quantities of biomass at reasonable prices and appropriate times. The task is challenging since, contrary to industrial logistics, the raw materials (oilseed and lignocellulosic crops) are produced slowly, seasonally, and with a limited yield, over vast territories. The paper proposes a Mixed Integer Linear Program (MILP) to optimize a multi-period and multi-biomass supply chain for several bio-refineries, at the tactical decision level. The locations of refineries can be fixed by the user or determined by the model. The aim is to minimize the total cost of the supply chain, including biomass production, pretreatments, storage, handling, bio-refineries setup and transportation, while satisfying given refinery demands in each period. The resulting MILP, already validated on medium-size instances, will be applied to a large-scale real case covering two regions of France (Champagne-Ardenne and Picardie) with 273 territorial units and 8 biomass types.

1 Introduction

Growing consciousness about destructive effects of climate change caused by greenhouse gas emissions, in addition to a huge rise in global demand for energy, have incitated many researchers to look for better alternatives to fossil fuels. Biofuel derived from biomass, as a renewable and clean energy source, is one of the few potential replacements of fossil fuels and can play a crucial role in the transition from traditional sources of energy.

N. Zandi Atashbar (✉) · N. Labadie · C. Prins
ICD-LOSI, University of Technology of Troyes (UTT), CS 42060, 10004 Troyes, France
e-mail: nasim.zandi_atashbar@utt.fr

N. Labadie
e-mail: nacima.labadie@utt.fr

C. Prins
e-mail: christian.prins@utt.fr

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_48

Biomass flow from field to fuel is called biomass supply chain and includes various activities such as cultivation, harvesting, handling, storage, transportation, and biofuel conversion. Although biomass itself is cheap relative to other sources of energy, its cost at refinery gates can be important, due to high logistics costs. That is why it is critical to improve the efficiency of its supply chain in order to make the production of biofuel affordable. Therefore, more and more researchers have been involved in modeling and optimizing biomass supply chains.

Designing the whole logistic system for a biomass supply chain has been always a tough challenge for researchers in this domain. Many studies focused on a single part or a few steps of the logistic system, e.g., a multi-biomass optimization model focusing on bioenergy conversion is proposed in [1]. A vast majority of papers consider a single type of biomass, see for instance the integrated optimization model presented in [2] to produce ethanol from switchgrass. Reference [3] considers several biomass types, like in our study, but for one pre-located biorefinery. A recent survey of optimization models for biomass supply chains can be found in [4–6].

To the best of our knowledge, this paper tackles for the first time a multi-period time horizon with different biomass types, centralized storages and several refineries which are already located or not. It describes a MILP to minimize a total cost, including biomass production, storage, handling, refineries setup and transport.

The paper is organized as follows: Sect. 2 presents the proposed model. The effectiveness of the model is illustrated with a numerical example in Sect. 3. Finally, concluding remarks are given in Sect. 4.

2 Data Description, Assumptions and Model

This research studies a comprehensive multi-period and multi-biomass supply chain with several node types. Biomass can be harvested in production zones, and then either stored in farm storages or transferred directly to centralized storages. Biomass can be also shipped from farm storages to centralized storages. Finally, it is transported to the refineries. The supply chain can be described by a graph G with a node-set N, composed of biomass production zones (subset BP), farm storages (FS), centralized storages (CS) and biorefinery input stocks (RL), and an arc-set A. Each arc $(i,j) \in A$ denotes a pre-computed shortest path from node i to node j, with length d_{ij} and a required vehicle type E_{ij}. A list of additional parameters and decision variables is given in Table 1.

A MILP model is proposed to minimize the total cost of this supply chain while satisfying given refinery demands in each period. The decision variables are biomass flows (from production zones to farm storages or centralized storages, from farm storages to centralized storages, and from farm storages and centralized storages to biorefineries), inventory levels of production zones, farm storages and centralized storages, and binary variables to locate biorefineries. The formulation (objective function and constraints) of the proposed model is described in the sequel.

Table 1 Notation

Variables	Unit	Description
S_i^t	Tonne	Stock of biomass at node i at end of time period t ($S_i^t \geq 0$)
F_{ij}^t	Tonne	Flow of biomass sent from node i to node j during time period t ($F_{ij}^t \geq 0$)
Y_z^r	–	1, if one biorefinery of type r is created in zone z; 0 otherwise ($Y_z^r \in \{0, 1\}$)
Parameters	Unit	Description
b_i	Period	First period when node i may be used
e_i	Period	Last period when node i may be used
κ_i	Tonne	Capacity of node i
ρ_i	–	Representative of the node i in case of shared storage capacity
S_i^b	Tonne	Initial inventory of node i in period b
S_i^e	Tonne	Minimum final inventory level of node i
C_i^S	€/(period × tonne)	Storage cost of node i
C_i^I	€/(period × tonne)	Input cost of node i (handling cost)
C_i^O	€/(period × tonne)	Output cost of node i (production cost for *BP* nodes, handling for others)
τ_i	Tonne/period	Maximum throughput of node i
δ_i	–	Degradation coefficient per period for node i, e.g., 0.999 for a 0.1% loss
C_v^V	€/(km × tonne)	Transportation cost for vehicle v
C_r^R	€	Setup cost of biorefinery type r
N_r	€	Number of refineries of type r
L_z	–	*Forbidden* if no refinery can be built in zone z, *Allowed*, if one may be created, or a *valid refinery type* if one refinery of this type is already installed

Objective function: The objective of this model is to minimize the total cost of biomass supply chain (*TC*), including biomass production, storage, biorefineries set-up, handling and transportation costs. *TC* includes the following terms.

$$CB = \sum_{i \in BP} \sum_{t \in [b_i, e_i]} \sum_{(i,j) \in A} C_i^O \times F_{ij}^t \tag{1}$$

$$CS = \sum_{i \in N \setminus BP} \sum_{t \in [b_i, e_i]} C_i^S \times S_i^t \tag{2}$$

$$CR = \sum_{z \in Z} \sum_{r \in R} C_r^R \times Y_z^r \tag{3}$$

$$CH = \sum_{i \in N \setminus BP} \sum_{t \in [b_i, e_i]} \sum_{(i,j) \in A} C_i^O \times F_{ij}^t + \sum_{i \in N} \sum_{t \in [b_i, e_i]} \sum_{(j,i) \in A} C_i^I \times F_{ji}^t \tag{4}$$

$$CT = \sum_{(i,j) \in A} \sum_{t \in [b_i, e_i]} d_{ij} \times C_{E_{ij}}^V \times F_{ij}^t \tag{5}$$

The mathematical model is:

$$\text{Min } TC = CB + CS + CR + CH + CT \tag{6}$$

$$\forall i \in N \mid \kappa_i < \infty \ \& \ \rho_i = 0, \ \forall t \in [b_i, e_i] : S_i^t \leq \kappa_i \tag{7}$$

$$\forall i \in N \mid \kappa_i < \infty \ \& \ \rho_i = i, \ \forall t \in [b_i, e_i] : \sum_{j \in N \mid \rho_j = i} S_j^t \leq \kappa_i \tag{8}$$

$$\forall i \in N \setminus RL \ \& \ S_i^e > 0 : S_i^{e_i} \geq S_i^e \tag{9}$$

$$\forall i \in N \setminus RL : S_i^b \times \delta_i + \sum_{(j,i) \in A} F_{ji}^{b_i} - \sum_{(i,j) \in A} F_{ij}^{b_i} = S_i^{b_i} \tag{10}$$

$$\forall i \in N \setminus RL, \ \forall t \in [b_i + 1, e_i] : S_i^{t-1} \times \delta_i + \sum_{(j,i) \in A} F_{ji}^t - \sum_{(i,j) \in A} F_{ij}^t = S_i^t \tag{11}$$

$$\forall i \in N \setminus RL \ \& \ \tau_i < \infty, \ \forall t \in [b_i, e_i] : \sum_{(i,j) \in A} F_{ij}^t \leq \tau_i \tag{12}$$

$$\forall z \in Z \mid L_z \neq Forbidden : \sum_r Y_z^r \leq 1 \tag{13}$$

$$\forall r \in R : \sum_{z \in Z \mid L_z \neq Forbidden} Y_z^r = N_r \tag{14}$$

$$\forall z \in Z \mid L_z \in R : Y_z^r = 1 \tag{15}$$

$$\forall i \in RL : S_i^b \times \delta_i \times Y_{NZ_i}^{NR_i} + \sum_{(j,i) \in A} F_{ji}^{b_i} - (D_{NP_i, NR_i}^{b_i} / \Delta_{NP_i}) \times Y_{NZ_i}^{NR_i} = S_i^{b_i} \tag{16}$$

$$\forall i \in RL, \ t \in [b_i + 1, e_i] : S_i^{t-1} \times \delta_i + \sum_{(j,i) \in A} F_{ji}^t - (D_{NP_i, NR_i}^t / \Delta_{NP_i}) \times Y_{NZ_i}^{NR_i} = S_i^t \tag{17}$$

$$\forall i \in RL \ \& \ S_i^e > 0 : S_i^{e_i} \geq S_i^e \times Y_{NZ_i}^{NR_i} \tag{18}$$

Equation (1) refers to the total production cost of biomass (cost of the outgoing flows from *BP* nodes). Equation (2) represents the total storage cost, not counted for *BP* nodes. Equation (3) is the total setup cost of refineries. Equation (4) represents the total handling cost. It includes the costs for loading products from any node except *BP* nodes, and the costs for unloading products at any node. Equation (5) shows the amount transported on each arc in each period multiplied by the distance and the cost per (tonne × km) of the vehicle specified for the arc. $E_{i,j}$ denotes the vehicle to be used on arc (i, j). Considering all the cost elements, the following objective function *TC* is obtained (which has to be minimized).

General constraints: Storage capacity constraints (7) and (8) apply to all nodes except the ones with an unlimited capacity, like *BP* nodes. A group of nodes *i* sharing the same storage capacity has a common representative ρ_i and we have to sum the stocks over all these nodes. Final inventory constraints (9) in the last period must be respected when they are specified, otherwise the constraints are redundant with $S_i^t \geq 0$. Constraints (10) and (11) guarantee the inventory balance for all nodes, except the *RL* nodes discussed in the next section. Constraints (10) are the particular case for the first period, with the initial inventory and constraints (11) correspond to the other periods. The maximum throughput constraints (12), when specified, are used to limit the total flow leaving an edge.

Constraints on refinery location: Constraints (13) ensure that at most one refinery can be built in each zone where creations are allowed. Constraints (14) guarantee that the number of refineries created for each type must be equal to the maximum number allowed. Constraints (15) force the set-up variable to 1 for an existing refinery (it is then eliminated by the pre-solver).

Constraints on *RL* nodes: The demand satisfaction constraints (16) and (17) are similar to the inventory balance equations (10) and (11) but the output flow is replaced by a demand in dry tonnes. Recall that NP_i and NR_i denote the product and the refinery type represented by node *i*. The need in dry tonnes in period *t* is $D^t_{NP_i,NR_i}$ and it must be divided by the percentage of dry matter Δ_{NP_i} to get the amount to be taken from the stock. Final inventory constraints (18) in the last period must be respected.

3 Numerical Example

The model is already tested on around 8000-km^2 area around the city of Compiègne (60 km North of Paris), with 29 zones (administrative districts containing several communes each and 1768 farms in total), 3 rape products (bulk seeds, straw bales and chaff bales), and a 1-year horizon divided into 52 weeks. Rape production in each zone was estimated using results of the 2010 Agricultural Census. The storage capacities and storage costs for silos (for seeds) and platforms (for bales) was obtained by sending a questionnaire to centralized storage operators. The costs of handling equipment and transport vehicles was found in professional databases. One refinery is assumed to be already located in Compiègne while a second may be created in any district with no common border with Compiègne.

The resulting instance was solved using Xpress-IVE 7.8 from FICO, on a 2.70 GHz Intel Core i7 portable PC with 32 GB of RAM and Windows 7 Professional. The model has 105,922 variables and 15,377 constraints. The pre-solver reduces it to 99,643 variables and 7,614 constraints. The relaxed LP is solved in 4.0 s to give a lower bound of 51.161×10^6. Then Xpress finds an optimal solution costing 52,057,533 € in 36.2 s. The cost of biomass represent 51.8%, capital and operating costs of refineries 38.4%, transport 5.1%, handling 2.6%, and storage 2.0%. By halv-

ing the demands, an optimal solution costing 35,355,177 € in 23.05 s is found. In addition, the model is tested when two refineries are assumed to be already located, one in Compiègne and the other one in Vic-sur-Aisne.The model has 99,590 variables and 7384 constraints. An optimal solution costing 52,515,225 € in 1.3 s is reached. Also, the possibility of creating and locating two other refineries in any district, has been tested. The model produced 99,645 variables and 7615 constraints. Xpress finds an optimal solution costing 52,057,225 € in 53.3 s. As demonstrated, by decreasing the number of binary variables related to locating biorefineries, the running time has been decreased significantly. If instead of predefined location of biorefineries, the model locates them, total cost will decrease. Also, when demands of biorefineries decrease, the total cost will decrease as well.

The running time is quite acceptable for a tactical model with binary setup variables. Moreover, the relaxed LP gives a very good lower bound. A project partner is preparing a large-scale instance covering two regions of France (Champagne-Ardenne and Picardie), with 273 districts and 8 biomass types.

4 Conclusion

In this paper, a mixed integer linear program is developed to optimize a multi-period and multi-biomass supply chain with several biorefineries. The objective is to minimize the costs of biomass production, storage, biorefineries set-up, handling and transport. The proposed mathematical formulation is general and flexible enough for adding new facilities and biomass products. It determines the amount of biomass produced, shipped and stored to satisfy demands of biorefineries during each period and the number, size and locations of biorefineries. Future research will focus on designing different solution approaches such as decomposition techniques, relaxation methods and meta-heuristics. Also, multi-modal transportation and multi-objective optimization are challenging issues toward which the research can be directed.

Acknowledgements This work was performed, in partnership with the SAS PIVERT, within the frame of the French Institute for the Energy Transition (Institut pour la Transition Energétique (ITE) PIVERT, http://www.institut-pivert.com) selected as an Investment for the Future (“Investissements d’Avenir”). This work was supported, as part of the Investments for the Future, by the French Government under the reference ANR-001-01.

References

1. Rentizelas, A.A., Tatsiopoulos, I.P.: Locating a bioenergy facility using a hybrid optimization method. Int. J. Prod. Econ. **123**, 196–209 (2010)
2. Zhang, J., Osmani, A., Awudu, I., Gonela, V.: An integrated optimization model for switchgrass-based bioethanol supply chain. Appl. Energy **102**, 1205–1217 (2013)
3. Ba, B.H., Prins, C., Prodhon, C.: A new tactical optimization model for bioenergy supply chain. In: International Conference on Applied Mathematics and OR, Vol. 2, No. 3, Miami, USA (2015)

4. Ba, B.H., Prins, C., Prodhon, C.: Models for optimization and performance evaluation of biomass supply chains: an OR perspective. Renew. Energy **87**, 977–989 (2016)
5. Gold, S., Seuring, S.: Supply chain and logistics issues of bio-energy production. J. Clean. Prod. **19**, 32–42 (2011)
6. De Meyer, A., Cattrysse, D., Rasinmaki, J., Van Orshoven, J.: Methods to optimise the design and management of biomass-for-bioenergy supply chains: a review. Renew. Sustain. Energy Rev. **31**, 657–670 (2014)

Transportation Planning with Different Forwarding Limitations

Mario Ziebuhr and Herbert Kopfer

Abstract In recent publications, it is assumed that there are transportation requests which have to be fulfilled by certain resources (e.g., own vehicle fleet or external carriers) due to contractual obligations. These requests are known as compulsory requests. The contribution of this publication is to identify the increase in transportation costs caused by a combination of compulsory requests with different contractual obligations. To evaluate the impact of compulsory requests, an existing column generation-based heuristic with two solution strategies for handling compulsory requests is applied and the generated results are analyzed.

1 Introduction

Today, forwarders are confronted with high demand fluctuations. That is why they have to reduce their costs and improve their flexibility by considering different fulfillment modes simultaneously. Beside their own transportation resources (self-fulfillment) forwarders use external carriers (subcontracting) and horizontal cooperation (collaborative planning) as fulfillment modes. One option of subcontracting is making use of a spot market where common carriers are employed for transportation requests in exchange of freight charges. A second option is the possibility of using long-term contractual agreements with subcontractors, where forwarders hire transportation capacities of long-term carriers to an agreed limit and take over the planning for the hired capacities. These subcontractors can be paid on a tour basis (TB) or on a daily basis (DB). Simultaneously solving the combined problem of vehicle routing for the private fleet and the optimal employment of common carriers and

M. Ziebuhr (✉) · H. Kopfer
Chair of Logistics, University of Bremen, Wilhelm-Herbst-Str. 5, 28359 Bremen, Germany
e-mail: ziebuhr@uni-bremen.de
URL:http://www.logistik.uni-bremen.de

H. Kopfer
e-mail: kopfer@uni-bremen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_49

subcontractors is known as integrated operational transportation planning (IOTP). Another fulfillment mode is given by collaborative transportation planning (CTP), where independent forwarders try to improve their planning situation by reallocating some of their requests or capacities within a horizontal coalition. The combination of an IOTP and CTP problem is denoted as collaborative operational transportation planning (COTP). In COTP, a request can be fulfilled either by self-fulfillment, subcontracting or collaborative planning.

In recent publications, it is assumed that some transportation requests cannot be fulfilled by certain fulfillment modes due to contractual obligations. These requests are denoted as compulsory [2, 4, 7, 8] or reserved requests [1]. In the following, the term compulsory requests is used. Respective publications consider extended pickup and delivery problems (PDPs); e.g., a pickup and delivery selection problem [1, 2, 4], an IOTP problem [7], and a COTP problem [8]. To solve these vehicle routing problems heuristic approaches are used; e.g., a memetic algorithm [4], a tabu-embedded simulated annealing algorithm [2], an adaptive large neighborhood search [1], and a column generation-based heuristic [7, 8].

This paper considers a COTP problem with forwarding limitations (COTPP-FL), which was introduced by [8]. In [8], compulsory requests with different contractual obligations are already considered. The contribution of this paper is to identify the increase in transportation costs by considering different kinds of compulsory requests simultaneously. To the best of our knowledge, there is no comparable approach in literature. To analyze the impact of compulsory requests, an existing column generation-based heuristic (CGB-heuristic) with two strategies for handling compulsory requests is used. In Sect. 2, a problem description is presented, while Sect. 3 describes the heuristic and Sect. 4 presents the computational studies.

2 Problem Description

This paper considers an IOTP problem, where a forwarder c hast to determine a transportation plan where n_c less than truckload requests have to be transported from their pickup $P_c = \{1, ..., n_c\}$ to their delivery location $D_c = \{n_c + 1, ..., 2n_c\}$. Thereby, the set of edges is defined by $A_c = V_c \times V_c$, while the set of nodes is given by $V_c = P_c \cup D_c \cup \{o_c\}$ where $\{o_c\}$ represents the depot. The distance d_{ij} is given for each edge $(i,j) \in A_c$. In IOTP, four fulfillment modes are applicable: private vehicles (K_c^1), rented vehicles based on mode TB (K_c^2), rented vehicles based on mode DB (K_c^3) and common carriers. A common carrier charges a fee γ_i for fulfilling the request at node i. Corresponding to objective function (1), the task is to determine a transportation plan which minimizes the sum of the fixed costs α_k, variable costs β_k, and fees γ_i by fulfilling routing, time, and loading constraints. Thereby, a transportation plan is defined by three binary decision variables: x_{ijk}, y_k^{DB}, and y_i^{CC}. The variable x_{ijk} is equal to one if a vehicle k travels from node i to j, y_k^{DB} is equal to one

if a rented vehicle $k \in K_c^3$ on mode DB is used, and y_i^{CC} is equal to one if a common carrier is employed for the fulfillment.

$$\min IP_c = \sum_{k \in K_1 \cup K_2} \sum_{(i,j) \in A_c} \beta_k d_{ij} x_{ijk} + \sum_{k \in K_c^3} a_k y_k^{DB} + \sum_{i \in P_c} \gamma_i y_i^{CC} \tag{1}$$

In terms of the COTP problem, this paper looks at the IOTP problem from a collaborative perspective, where forwarders align their individual transportation plans by exchanging requests with each other. Depending on the transportation plans of the coalition members, each forwarder c offers a request portfolio P_c^- for exchange and receives a new portfolio P_c^+ after the request exchange process is completed. The offered request portfolio P_c^- is defined by $P_c \backslash P_c^0$, where P_c^0 represents the set of non-transferable requests. As soon as the request exchange process is completed, a forwarder c is responsible for producing the assigned request portfolio P'_c with $P'_c = P_c^0 \cup P_c^+$ and fulfillment costs defined by IP'_c. The goal of the COTP problem is to minimize the individual costs of each member of the coalition (Eq. (2)) by ensuring that each exchanged request is fulfilled by exactly one coalition member (Eq. (3)) and that all offered requests are assigned to the coalition members (Eq. (4)).

$$\min CTP_c = \sum_{c=1}^{m} IP'_c \tag{2}$$

$$P'_c \cap P'_h = \emptyset, \qquad \forall c, h = 1, ..., m,\ c \neq h, \tag{3}$$

$$\cup_{c=1}^{m} P_c^- = \cup_{c=1}^{m} P_c^+. \tag{4}$$

The described COTP problem does not consider compulsory requests. Therefore, it is necessary to extend the COTP problem. In the proposed COTPP-FL, four different types of compulsory requests are considered which differ in terms of the applicable external resources for fulfilling these requests. The following request types are considered: S requests (fulfillment by any fulfillment mode), P1 requests (fulfillment by self-fulfillment), P2 requests (fulfillment by self-fulfillment and long-term carrier), P3 requests (fulfillment by self-fulfillment and collaboration), and P4 requests (fulfillment by self-fulfillment, long-term carrier, and collaboration). It means for example, a P3 request requires the application of a vehicle of the private fleet of any member within the horizontal coalition, while the application of subcontracting is strictly prohibited. All P1-P4 requests are compulsory requests and have a common feature of being not able to use common carriers. In terms of the mathematical formulation of the COTPP-FL, the set of pickup nodes P_c has to be separated into five disjoint sets: S pickup nodes S_c, P1 pickup nodes P_c^1, P2 pickup nodes P_c^2, P3 pickup nodes P_c^3, and P4 pickup nodes P_c^4. Based on these disjoint sets, the COTP problem has to be extended by constraints which ensure that just the applicable resources are used for compulsory requests. A COTPP-FL is presented by [8].

3 Solution Approach

As solution approach, the CGB-heuristic which was introduced by [6], is applied and modified. In the CGB-heuristic, it is proposed to reformulate the transportation problem into two problems: the master problem (selection of vehicle routes) and the subproblem (generation of vehicle routes). Thereby, the subproblem is solved by each coalition member separately while the master problem is solved by a neutral software agent. By solving the master problem dual values are generated and forwarded to the subproblem for identifying new vehicle routes which reduce the operational costs. The process is repeated for a certain number of iterations. The subproblem is solved by an adaptive large neighborhood search (ALNS) and the master problem is solved by a commercial solver. An ALNS, which was introduced by [3], is a local search heuristic which uses a simulated annealing and different removal and insertion heuristics. The CGB-heuristic is executed two times. First, the approach is applied to the complete request portfolio P which means that each coalition member is able to bid on all requests of the coalition. Secondly, each coalition member uses this approach for the winning bids of the first application of the CGB-heuristic.

To consider compulsory requests two strategies are proposed, which are explained in detail by [7]. One strategy for handling compulsory requests is the strict generation procedure where the compulsiveness of requests is strictly observed by the subproblem. Thereby, only valid vehicle routes are accepted by the simulated annealing during the ALNS. This means that a request is fulfilled by a fulfillment mode corresponding to the request type. To generate as many valid vehicle routes as possible, three modifications are recommended for the ALNS. First, the common carrier option is penalized by using penalty costs for compulsory requests. Second, the fulfillment of P1 and P2 requests by different coalition members as well as the fulfillment of P1 and P3 requests by long-term carriers are prohibited by skipping these invalid insertions for the insertion heuristics. Third, the request portfolio of the insertion heuristics is split into two: one for compulsory requests and one for standard requests. Due to this procedure compulsory requests are preferred for reinsertion. A second strategy is the strict composition procedure where forwarding limitations are ignored by the ALNS and considered by the solution of the master problem. It means that many of the submitted routes may contain compulsory requests which are served by an improper fulfillment mode. To ensure that feasible vehicle routes are selected for the considered COTPP-FL, the master problem is extended by new constraints which observe the applied fulfillment modes for compulsory requests. Therefore, the vehicles are numbered in an ascending order and for each compulsory request a certain range is determined and ensured. To ensure feasible solutions by high ratios of compulsory requests, it is proposed that the ALNS accepts only feasible vehicle routes in the first round of the column generation.

4 Computational Experiments

In this paper, the instances of [8] are used which are derived based on 24 COTP instances where 2–5 IOTP instances with the same location structure (R1, C1, and RC1) are combined to one COTP instance. In general, the instances are derived from the well known instances of [5]. The size of the available fleet size is set to the number of vehicles used in the best-known PDP solutions. The total vehicle fleet of each coalition member is composed of: 40% private vehicles, 30% vehicles on mode TB, and 30% vehicles on mode DB. In this paper, the instances of [8] with 15% compulsory requests and 85% standard requests are considered. Regarding these 15% compulsory requests, eleven different combinations are analyzed: (P1, P2), (P1, P3), (P1, P4), (P2, P3), (P2, P4), (P3, P4), (P1, P2, P3), (P1, P2, P4), (P1, P3, P4), (P2, P3, P4), and (P1, P2, P3, P4). In the existing instances, the type of compulsory request is not defined. That it is why, first, the number of compulsory requests in an instance is divided by the number of request types in a combination. Then, each request type receives the same number of requests in an ascending order based on the existing request order. In case that an equal distribution is not available, the last request type within a combination receives the remaining compulsory requests. 15 samples are generated for each instance and combination. The same parameter setting is used as suggested by [8]. As evaluation criterion, the percentaged cost increase between the COTPP-FL solution computed by our best heuristic and the best-known COTP solution is used. The experiments are executed on a Windows 7 PC with Intel Core i7-2600 processor (3.4 GHz and 16 GB of memory) and the solver CPLEX (version 12.51) is applied.

In a first study, the strict generation procedure is compared with the strict composition procedure for all combinations and instances with three coalition members. Thereby, it is worth mentioning that in [8] it is identified that the strict generation procedure is preferable for P1 and P3 requests, while the strict composition procedure is preferable for P2 and P4 requests. Corresponding to this observation, it is assumed that the strict generation procedure is preferable when most of the compulsory requests are P1 and P3 requests, while the strict composition procedure is preferable in combinations with P2 and P4 requests. In our study, this assumption can often be verified by identifying that every time when P3 requests are considered the strict generation procedure leads to better results.

In a second study, the increase in transportation costs is analyzed by considering different combinations of compulsory requests. Thereby, the COTPP-FL is solved by the CGB-heuristic combined with the best strategy for each combination. The percentaged increase in costs per compulsory request as well as the best COTP solution are presented in Table 1. To evaluate the results of Table 1 it is necessary to know the findings of [8] where it is observed that on average P1 requests lead to the highest, P3 to the second highest, P2 to the third highest, and P4 requests to the forth highest additional costs. Corresponding to this finding, the following order in terms of additional costs is expected for combinations with two request types: (P1, P3), (P1, P2), (P2, P3), (P1, P4), (P3, P4), and (P2, P4) and for combinations with three

Table 1 Percentaged increase in costs per compulsory request

Instance			Best COTP	Combinations (cost increase in %)										
id	m	n		(1,2)	(1,3)	(1,4)	(2,3)	(2,4)	(3,4)	(1,2,3)	(1,2,4)	(1,3,4)	(2,3,4)	(1,2,3,4)
C101	2	105	5348	0.686	1.069	0.618	0.572	0.208	0.555	0.771	0.543	0.803	0.479	0.659
C102	2	106	5340	0.615	0.884	0.497	0.381	0.068	0.482	0.661	0.471	0.683	0.352	0.513
C103	3	159	7910	0.440	0.632	0.344	0.346	0.143	0.316	0.497	0.332	0.481	0.297	0.388
C104	3	159	7335	0.457	0.678	0.355	0.347	0.117	0.330	0.538	0.360	0.522	0.262	0.424
C105	4	212	9912	0.357	0.442	0.212	0.220	0.089	0.195	0.323	0.226	0.309	0.196	0.249
C106	4	211	9721	0.459	0.592	0.269	0.346	0.140	0.263	0.439	0.338	0.407	0.272	0.356
C107	5	264	12296	0.314	0.440	0.187	0.224	0.082	0.193	0.299	0.211	0.292	0.185	0.229
C108	5	264	12386	0.364	0.498	0.184	0.242	0.071	0.209	0.331	0.214	0.345	0.186	0.238
R101	2	104	7052	0.304	0.400	0.209	0.269	0.124	0.167	0.347	0.227	0.238	0.179	0.227
R102	2	104	7504	0.375	0.781	0.309	0.362	0.068	0.265	0.431	0.272	0.459	0.276	0.329
R103	3	160	10678	0.287	0.379	0.220	0.267	0.138	0.195	0.311	0.222	0.292	0.217	0.253
R104	3	154	9854	0.273	0.463	0.209	0.265	0.091	0.181	0.342	0.227	0.297	0.176	0.238
R105	4	208	13333	0.247	0.417	0.181	0.218	0.116	0.152	0.260	0.200	0.277	0.169	0.205
R106	4	215	15508	0.189	0.255	0.122	0.160	0.081	0.093	0.177	0.140	0.156	0.117	0.140
R107	5	265	16222	0.211	0.264	0.192	0.177	0.103	0.116	0.219	0.211	0.176	0.149	0.164
R108	5	262	16908	0.181	0.286	0.171	0.166	0.101	0.115	0.188	0.158	0.185	0.126	0.144
RC101	2	106	7415	0.348	0.552	0.285	0.315	0.197	0.264	0.372	0.283	0.374	0.247	0.307
RC102	2	107	8058	0.320	0.544	0.247	0.313	0.179	0.246	0.370	0.286	0.335	0.232	0.300
RC103	3	160	10801	0.267	0.374	0.209	0.280	0.163	0.199	0.325	0.229	0.286	0.249	0.275
RC104	3	161	11485	0.312	0.441	0.221	0.333	0.167	0.195	0.374	0.255	0.301	0.265	0.277
RC105	4	211	13783	0.286	0.394	0.212	0.263	0.181	0.196	0.282	0.235	0.273	0.216	0.241
RC106	4	213	14711	0.237	0.346	0.188	0.222	0.157	0.163	0.241	0.203	0.247	0.188	0.203
RC107	5	265	17231	0.249	0.354	0.172	0.228	0.142	0.152	0.250	0.197	0.236	0.183	0.203
RC108	5	266	18339	0.238	0.311	0.155	0.219	0.145	0.163	0.223	0.196	0.233	0.191	0.200

request types: (P1, P2, P3), (P1, P3, P4), (P1, P2, P4), and (P2, P3, P4) (descending order). This assumption is verified in the second study. Further on, it is also obvious that the highest additional costs over all combinations can always be observed for the combination (P1, P3), while the lowest ones can always be observed for the combination (P2, P4). The remaining combinations differ a little bit more regarding their additional costs depending on the location structure and the number of freight forwarders. It is also observed that there is no significant cost reduction by considering different kinds of compulsory requests simultaneously. Since the identified figures are just slightly lower (on average about 7%) than the aggregated figures of [8], it is assumed that the procedure for distributing different compulsory requests within the instance generation is responsible for this observation.

Acknowledgements This research was supported by the German Research Foundation (DFG) as part of the project "Kooperative Rundreiseplanung bei rollierender Planung".

References

1. Li, Y., Chen, H., Prins, C.: Adaptive large neighborhood search for the pickup and delivery problem with time windows, profits, and reserved requests. Eur. J. Oper. Res. **252**(1), 27–38 (2016)
2. Ramaekers, K., Caris, A., Maes, T., Janssens, G.: Pickup and delivery selection problem formulation and extension to problem variants. Inf. Technol. Manag. Sci. **18**, 84–90 (2016)
3. Ropke, S., Pisinger, D.: An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp. Sci. **40**(4), 455–472 (2006)
4. Schönberger, J.: Operational Freight Carrier Planning. Springer, Berlin (2005)
5. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. **35**(2), 254–265 (1987)
6. Wang, X., Kopfer, H., Gendreau, M.: Operational transportation planning of freight forwarding companies in horizontal coalitions. Eur. J. Oper. Res. **237**(3), 1133–1141 (2014)
7. Ziebuhr, M., Kopfer, H.: Solving an integrated operational transportation planning problem with forwarding limitations. Transp. Res. Part E: Logistics Transp. Rev. **87**, 149–166 (2016)
8. Ziebuhr, M., Kopfer, H.: Transportation planning with forwarding limitations. In: Freitag, M., Kotzab, H., Pannek, J. (eds.) Dynamics in Logistics—Proceedings of the 5th International Conference LDIC, pp. 225–234. Springer (2016)

Part XII
Metaheuristics

Alternative Fitness Functions in the Development of Models for Prediction of Patient Recruitment in Multicentre Clinical Trials

Gilyana Borlikova, Michael Phillips, Louis Smith, Miguel Nicolau and Michael O'Neill

Abstract For a drug to be approved for human use, its safety and efficacy need to be evidenced through clinical trials. At present, patient recruitment is a major bottleneck in conducting clinical trials. Pharma and contract research organisations (CRO) are actively looking into optimisation of different aspects of patient recruitment. One of the avenues to approach this business problem is to improve the quality of selection of investigators/sites at the start of a trial. This study builds upon previous work that used Grammatical Evolution (GE) to evolve classification models to predict the future patient enrolment performance of investigators/sites considered for a trial. Selection of investigators/sites, depending on the business context, could benefit from the use of either especially conservative or more liberal predictive models. To address this business need, decision-tree type classifiers were evolved utilising different fitness functions to drive GE. The functions compared were classical accuracy, balanced accuracy and F-measure with different values of parameter beta. The issue of models' generalisability was addressed by introduction of a validation procedure. The predictive power of the resultant GE-evolved models on the test set was compared with performance of a range of machine learning algorithms widely used for classification. The results of the study demonstrate that flexibility of GE induced classification models can be used to address business needs in the area of patient recruitment in clinical trials.

G. Borlikova (✉) · M. Nicolau · M. O'Neill
Natural Computing Research & Applications Group, School of Business, University College Dublin, Dublin, Ireland
e-mail: gilyana.borlikova@ucd.ie

M. Nicolau
e-mail: miguel.nicolau@ucd.ie

M. O'Neill
e-mail: m.oneill@ucd.ie

M. Phillips · L. Smith
ICON plc, Dublin, Ireland
e-mail: michael.phillips@iconplc.com

L. Smith
e-mail: louis.smith@iconplc.com

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_50

1 Introduction

For any drug to be approved for human use, its safety and efficacy need to be evidenced through clinical trials. Patient recruitment is the most time and resource consuming part of the majority of clinical trials. One of the avenues to approach this business problem is to improve the quality of selection of investigators/clinical sites (sites) at the start of a trial. This study builds upon previous work [2] that used Grammatical Evolution (GE) [4, 10], a grammar-based Genetic Programming system [9] to evolve classification models to predict the future patient enrolment performance of sites considered for a trial. Development of predictive models needs to take into account the business context in which the models will be deployed. Misclassification costs will inevitably differ depending on a particular business situation (i.e. abundance or scarcity of the eligible sites, site setup costs, penalties for the missed timelines, etc.). Therefore, different business contexts might benefit from the use of either especially conservative or more liberal predictive models. This study evolved decision-tree type classifiers using different fitness functions to drive GE. The results demonstrate that utilisation of different fitness functions can be used to address challenges of uneven misclassification costs in unbalanced data situation and guide evolution of customised patient recruitment classification models by GE.

2 Problem Definition and Background

Notwithstanding some recent developments and new approaches [13] to patient recruitment it remains an area of active business interest. Developing tools for more robust site selection at the beginning of a trial can help avoid delays and reduce the need for "rescue" sites in the course of the trial. Predictive business analytic techniques can be used to improve this process, such as development of site classification models based on the historic data. However, model construction for this problem requires care as in most real-life patient recruitment situations historic data is unbalanced (the proportion of successful vs. poorly performing sites is uneven) and, very often, the costs of misclassification error for different classes are different. One type of error (False Negative, FN, in terms of Confusion Matrix) will result in inclusion of a potentially weak site in the study, while another (False Positive, FP) will lead to exclusion of a potentially promising site from the study. [3, 6, 12] advocate the use of ROC and AUC for model evaluation and selection and the use of expected cost/benefit to frame classifier evaluation especially in the context of probabilistic machine learning (ML) classifiers. In the GP field Zhang and colleagues [1] developed different fitness functions to improve classification in the case of unbalanced data. Cost-sensitive learning is an active area of research following [5]. In this study we follow up on our previous work that adopted decision-tree type GE classifiers with discreet class output to address prediction in patient recruitment. We investigated the use of different fitness functions to drive GE: classical accuracy, balanced accuracy and F-measure with different values of parameter beta [7, 11, 14].

3 Experiments, Results and Analysis

The dataset used was described previously in [2] and constructed based on the historical data provided by ICON plc. on 21 Diabetes Mellitus Type II Phase III clinical trials. The prepared dataset consisted of 1233 records and 42 predictor variables (35 numerical and 7 categorical) describing different aspects of investigator/site. All sites were divided into two classes based on their patient recruitment performance. GE was used to evolve decision-tree type discreet classifiers. The GE grammar (similar to the previous work) used the function and terminal set detailed in Table 1. The evolutionary parameters were set as follows: population size 1000 individuals, 50 generations, ramped-half-and-half initialisation, tournament selection, tournament size 5, generational replacement, elite size 1, sub-tree crossover (90% probability), sub-tree mutation (1 event per individual), maximum derivation tree depth was 9, 30 independent runs.

The data was split into train and test subsets (balanced, 70/30%) and model training and tuning was performed using the train subset. Performance of the best of run models was then tested on the test subset to ascertain how well they generalise to unseen data. Models were evolved on 5 different splits of data (data cuts 1–5). Selected models were evaluated on 5 different test subsets in order to further assess models' generalisation ("Monte-Carlo cross-validation", [8]). The resultant generalisation metrics have some inbuilt optimistic bias, but are still better than assessment on a single test subset. The following functions were investigated [7, 14]:

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)}; \tag{1}$$

$$Balanced\ Accuracy = \omega \times \frac{TP}{TP + FN} + (1 - \omega) \times \frac{TN}{TN + FP};\ where\ \omega = 0.5 \tag{2}$$

$$F_{\beta} measure = (1 + \beta^2) \times \frac{TP}{((1 + \beta^2) * TP + \beta^2 \times FN + FP)};\ where\ \beta = 0.5, 1, 2 \tag{3}$$

Performance of the GE models was compared to a number of well-established ML algorithms. The R CARET package [8] was used to train, tune and test the ML models. Accuracy was used as a metric in the ML cross-validation procedure and all

Table 1 Function and terminal sets of GE classifier

Function set	Terminal set
+, −, *, /, and, or, not	35 numerical predictive variables: x0, ..., x34
equals, not_equals	3 categorical predictive variables: x35, x36, x37
less, greater, less_e, greater_e	4 Boolean predictive variables: x38, ..., x42
	20 random constants in −1.0, ..., 1.0 with 0.1 step

Table 2 Benchmark machine learning (ML) model settings

Model	R CARET method	Parameter setting
Support Vector Machines, Radial Basis Function Kernel (**svm**)	svm	sigma = 0.0149, cost = 0.5
Classification and Regression Tree (**cart**)	rpart	complexity parmeter = 0.0249
Multivariate Adaptive Regression Splines (**mars**)	gcvEarth	product degree = 1
Random Forest (**rf**)	rf	#randomly selected predictors = 7
Nearest Shrunken Centroids (**nsc**)	pam	shrinkage threshold = 4.1236

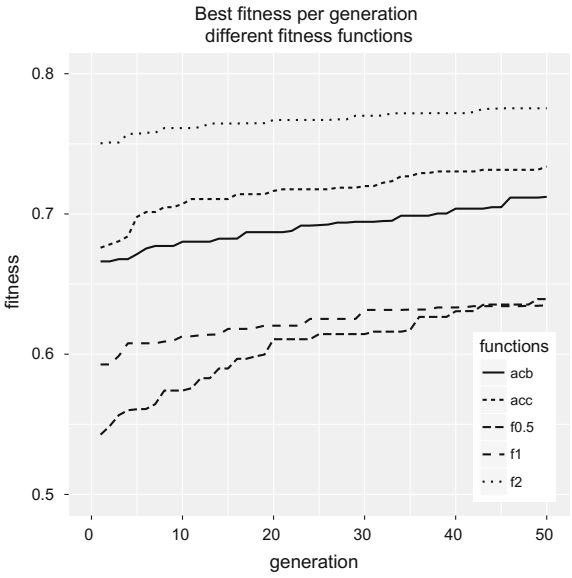


Fig. 1 Best fitness achieved by GE models driven by different fitness functions during training on data cut 1 over 50 generations

ML models used default class probability threshold of 0.5. ML models' settings are presented in Table 2.

The best (Fig. 1) and average population fitness gradually increased over 50 GE generations in all experiments confirming the ability of all used fitness functions to successfully drive evolutionary process.

As the next step, best of run individuals were assessed on the previously unseen test subset. To facilitate between-function comparison of performance of all models, models were assessed in terms of True Positive Rate (TPR) and False Positive Rate (FPR) coordinates (TPR = TP/Condition Positive, FPR = FP/Condition Negative). Dot-plots of 30 individual runs with each fitness function (Fig. 2) show that different functions evolve models that reside in the different parts of the TPR-FPR space. Best model evolved on data cut 1 by accuracy achieved 0.57/0.27, by balanced accuracy −0.77/0.45, by $F_{0.5}$ − 0.61/0.30, by F_1 − 0.84/0.52, by F_2 − 0.99/0.79 TPR/FPR

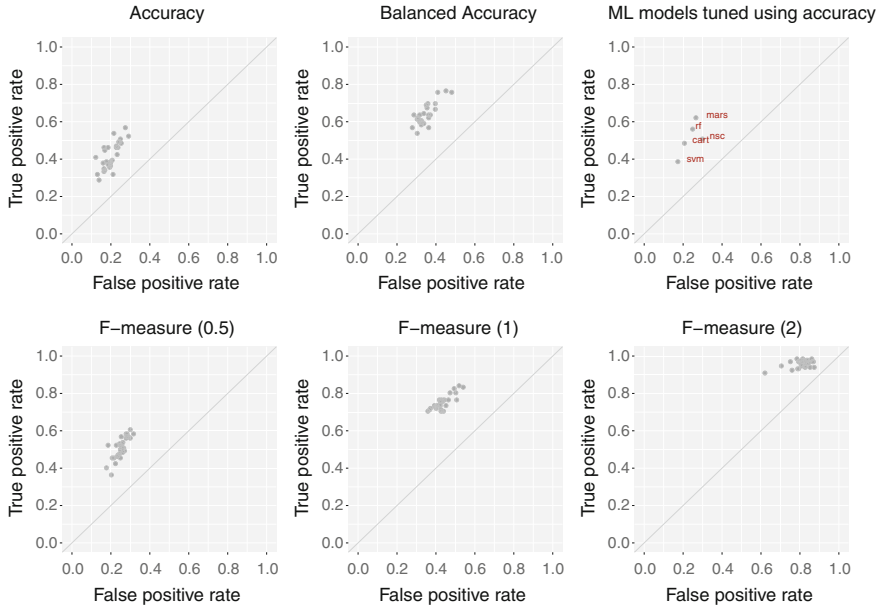


Fig. 2 Performance of the best of run models evolved using different fitness functions on test subset of data cut 1 (30 runs per fitness function); ML models with 0.5 class threshold probability

respectively. For comparison, the best in terms of TPR ML model in this experiment MARS demonstrated 0.62/0.27 TPR/FPR. The results clearly demonstrate that GE evolves models that are comparable or even better than ML evolved models, depending on the context (Fig. 2).

When assessed for generalisation across 5 different test subsets all models demonstrated steady performance (Fig. 3). For each of the fitness functions the top classifiers evolved on the five training data-cuts were evaluated on the five test subsets and the resultant averaged performance was used to visualise the "gross" generalisability of classifiers from each function. Last pane of Fig. 3 illustrates the results. The figure shows that, unsurprisingly, the increase in TPR comes at the price of increased FPR. In the extreme case classifiers evolved using F2-measure achieve 0.99 TPR, but at the cost of 0.8% FPR.

Results of this study clearly demonstrate that the use of different fitness functions to drive GE successfully evolves models positioned in different parts of the TPR/FPR space. It is important to note that while the use of the investigated functions does not produce "principally better" models on this challenging dataset; it allows to develop models that target different parts of TPR/FPR space, akin to the custom choice of classification threshold in case of ML algorithms that return class probabilities. This variety of models can be exploited in the real-world business situation to select models that are suited to particular business circumstances.

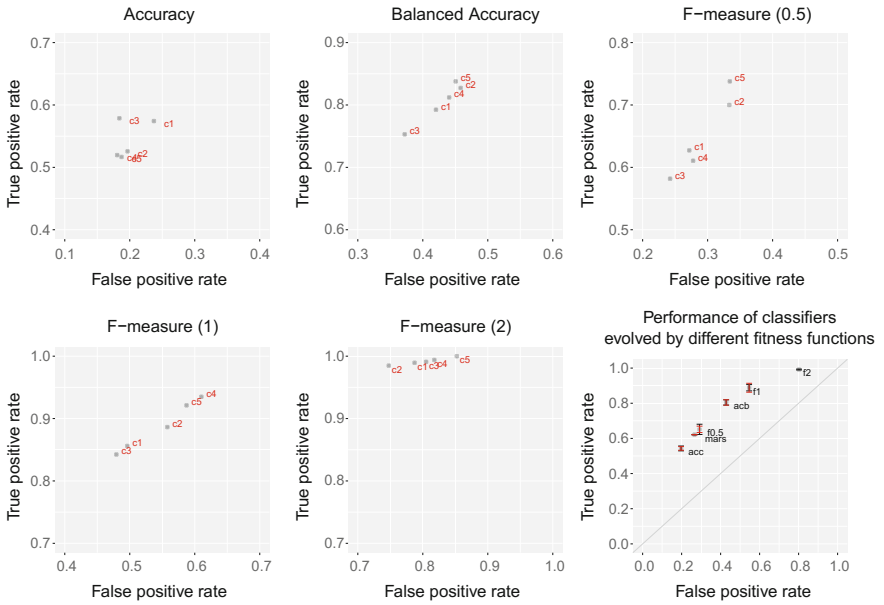


Fig. 3 Generalisation between different test subsets of the top classifiers evolved on 5 training subsets (c1–c5) using different fitness functions. *Last panel*—Gross averaged performance of classifiers evolved using different fitness functions (mean sem, *black error-bars*—sem on tpr, *red*—-sem on fpr, *mars—evaluation of MARS ML model on data-cut 1)

4 Conclusions

This paper approached the business problem of improving patient recruitment for clinical trials by developing models to predict future performance of clinical sites. The GE-based classifiers were evolved using accuracy, balanced accuracy and F-measure as fitness functions with the aim to produce GE classifiers with different true positive rate/false positive rate qualities. The results demonstrate that utilisation of different fitness functions can be used to address challenges of uneven misclassification costs in unbalanced data situation and guide evolution of customised patient recruitment classification models by GE. However, a more detailed business assessment of misclassification costs in each case is needed to allow for the full quantisation of the models' performance in the business sense.

Acknowledgements The authors would like to thank Dr. Michael Fenton from the UCD Natural Computing Research and Applications Group for his insightful advice on GE methodology. This research is based upon work supported by ICON plc.

References

1. Bhowan, U., Johnston, M., Zhang, M.: Developing new fitness functions in genetic programming for classification with unbalanced data. IEEE Trans. Syst. Man Cybern. Part B (Cybern.) **42**(2), 406–421 (2012)
2. Borlikova, G., Phillips, M., Smith, L., O'Neill, M.: Evolving classification models for prediction of patient recruitment in multicentre clinical trials using grammatical evolution. In: EvoApp's 2016, pp. 46–57. Springer (2016)
3. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognit. **30**(7), 1145–1159 (1997)
4. Dempsey, I., O'Neill, M., Brabazon, A.: Foundations in Grammatical Evolution for Dynamic Environments, Studies in Computational Intelligence, vol. 194. Springer (2009)
5. Elkan, C.: The foundations of cost-sensitive learning. In: Proceedings of the Seventeenth International Joint Conference of Artificial Intelligence, pp. 973–978. Seattle, Washington (2001)
6. Fawcett, T.: An introduction to ROC analysis. Pattern Recognit. Lett. **27**(8), 861–874 (2006)
7. Ferri, C., Hernandez-Orallo, J., Modroiu, R.: An experimental comparison of performance measures for classification. Pattern Recognit. Lett. **30**(1), 27–38 (2009)
8. Kuhn, M., Johnson, K.: Applied Predictive Modeling. Springer, New York (2013)
9. McKay, R.I., Hoai, N.X., Whigham, P.A., Shan, Y., O'Neill, M.: Grammar-based genetic programming: a survey. Genet. Program Evolvable Mach. **11**(3/4), 365–396 (2010)
10. O'Neill, M., Ryan, C.: Grammatical Evolution: Evolutionary Automatic Programming in a Arbitrary Language, Genetic programming, vol. 4. Kluwer Academic Publishers (2003)
11. Powers, D.M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation (2011)
12. Provost, F., Fawcett, T.: Data Science for Business: What you Need to Know About Data Mining and Data-Analytic Thinking. O'Reilly Media, Inc. (2013)
13. Schuler, P., Buckley, B.: Re-engineering Clinical Trials: Best Practices for Streamlining the Development Process. Academic Press (2014)
14. Sokolova, M., Japkowicz, N., Szpakowicz, S.: Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation. In: Australasian Joint Conference on Artificial Intelligence, pp. 1015–1021. Springer (2006)

Long-Term Consequences of Depot Decisions for the Inventory Routing Problem

Sandra Huber and Martin Josef Geiger

Abstract The article describes our work on an extension of the Inventory Routing Problem (IRP). While the basic formulation of the IRP combines delivery volume with vehicle routing decisions, and thus includes tactical and operative aspects, we here consider the strategic placement of the depot, also. In some first experiments, the effect of such a strategic decision on the inventory levels and routing costs in the supply chain is studied. Besides, potential improvements, i.e., savings in transportation costs are investigated. Our findings indicate that efficiency improvements are indeed achievable by extending the problem formulation towards the strategic planning level.

1 Introduction and Problem Statement

Facility location problems, vehicle routing problems and inventory management are key problem areas in supply chain management [11]. The bi-objective IRP incorporates decisions on a tactical as well as on an operational planning level, such as the combination of inventory management with a Capacitated Vehicle Routing Problem (CVRP). For a detailed IRP literature review we refer to the work of [3, 4]. Furthermore, it seems promising to include a strategic planning level in the IRP to analyze the tradeoff between the two proposed objectives. Without the integration of the depot decision, previous studies show that the two objectives are clearly in conflict to each other: the simultaneously minimization of the total sum of all inventory levels at each customer at the end of each period and the total sum of all distances traveled by the vehicles in each period. While small delivery quantities lead to low inventory levels over time, large delivery quantities allow a minimization of the routing costs [5, 8]. The surveys of [6, 7, 9] investigate the combined location routing

S. Huber (✉) · M.J. Geiger
Helmut-Schmidt University, Holstenhofweg 85, 22043 Hamburg, Germany
e-mail: sandra-huber@hsu-hh.de

M.J. Geiger
e-mail: m.j.geiger@hsu-hh.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_51

and inventory problem. Choosing depots from several potential locations and determining routes to meet customers' demands is the aim of these works [9].

We propose a single-item IRP with repeated deliveries in a distribution network with one depot and a geographically dispersed set of n customers over a finite planning horizon. Inventory costs and capacities are considered at the customers, but not at the depot. Also, the number of vehicles is unconstrained and capacitated vehicles are used [5]. A deterministic IRP is assumed where the consumption for each customer and each period is known beforehand. When the current inventory is insufficient to satisfy the forthcoming demands d_{it} of a customer i at each period t, an action (delivery) of the supplier is needed. For satisfying the demand at the customers, we consider the strategy that the inventory currently held at the customers is either able to fully cover the customers demands at period t or the inventory is zero. Following this idea, our replenishment strategy is to avoid stockout situations by shipping enough goods in advance or just in time. With respect to the data, the customers demand can vary from period to period, resulting in changing delivery quantities over the time horizon T (dynamic IRP [8]).

This IRP model description must deal with the following decisions: (1) the depot location must be selected for the whole planning horizon, (2) delivery quantities q_{it} for each customer i, $i = 1, \ldots, n$ and each period $t \in T$ must be determined, and (3) the VRP must be solved for each period t, $t = 1, \ldots, T$ including the delivery quantities q_{it} into tours for the involved vehicles with the already selected depot.

Two objectives are considered for the IRP: The total sum of inventory levels $f_1 = \sum_{t=1}^{T} \sum_{i=1}^{n} L_i^t$ is minimized, where L_i^t is the inventory level for customer i for period t. This objective function $f_2 = \sum_{t=1}^{T} \mathrm{VRP}_t(q_{it}, \ldots, q_{nT})$ expresses the minimization of the total sum of the routing costs [5]. An overview of models with different objectives is described in [2].

2 Solution Method

Our solution method separates the problem into two decision levels: (1) the determination of delivery volumes and (2) the subsequent computation of the routing for each period which takes into account the previous calculated delivery quantities. In terms of the delivery strategy, alternatives are encoded by a n-dimensional vector $\pi = (\pi_1, \ldots, \pi_n)$ of integers. Each element π_i corresponds to a customer i and reflects for how many periods the demand of customer $i, i = 1, \ldots, n$ is covered (delivery period). A delivery takes place when the demand cannot be satisfied with the inventory level at the customer.

We define an initial solution by identical periods for all customers, starting with 1 and increasing them by steps of 1 until the alternative cannot be added to the archive of non-dominated solutions. This is e.g. due to the fact that capacity constraints of the vehicles and at the customers must be met. For example, when the delivery period is defined as $\pi_i = 4$, then the exact demand of customer i is satisfied for the next four

consecutive periods. When the 'delivery period' is set to four for every customer, it is called 'identical delivery periods'.

Based on this delivery strategy, the supplier ensures that the customer does not run out of products. Also, a growing demand over the time horizon results in higher delivery quantities. This is a rather direct idea of representing delivery policies. Alternatively, a 'constant-delivery-quantity-approach' could be applied, which leads to changing delivery periods. A more general solution is provided in [1], where customers are synchronized.

To improve the initial solutions, a run of the local search is performed on the n-dimensional vector π which represents the delivery periods. Particularly, an algorithm is used to change every value within π by ± 1. Values < 1 are avoided since we assume 1 as the smallest measurement of a period and a customer cannot be delivered twice a day. Throughout the search, an unbounded archive is kept which deletes solutions by dominance comparisons. Previous results indicate that the memory of a typical computer is sufficient to store these solutions [8].

3 Computational Experiments

Experiments are carried out to investigate the performance of the integration of depot decisions. In particular, two research questions are raised: (1) To which extent does the choice of the depot influence the long-term consequences of the subsequent vehicle routing and the inventory management? (2) Can we, even in rather straightforward experiments, find improved depot locations and if so, what is the magnitude of the improvement?

The instances are proposed by [10] and usually the depot is positioned in the 'center' of the network so-called 'original depot'. Here, the x-coordinate is 30 and the y-coordinate is 40. To get an idea of the graph, the x-coordinates range from a minimal value of 5 to a maximum value of 63 and the y-coordinates lie in the range from 6 to 69.

From a strategic planning level viewpoint, it might be beneficial to relocate the depot at some future time. Thus, we compare the 'original depot' with other depot locations. We assume that the possible depot location is at the customer's site. For example, when an instance has 50 customers, 50 different depot locations can be analyzed. Since previous studies achieved similar results for different instances [8], we here restrict the presentation on *GS-b-01* with 30 periods and 50 customers. Demand data of the customers increase over time. With respect to every period, the demand values can vary by $\pm 25\%$ around the average demand. In order to verify a similar behaviour of the algorithm, further test runs for different data sets must be performed. Note that all experiments are executed on a single core of an Intel Core i7-6600U CPU 2.60 GHz with 8 GB RAM. Also, the maximal number of evaluations is set to 300,000. Note that it takes 177 min to compute the approximation of the Pareto front for the 'original depot' (*GS-b-01*). In each evaluation, 30 vehicle routing problems have to be solved, so the computation time is considerably higher in comparison to

what we commonly find in the literature. However, an earlier termination is possible when the local search has already investigated all alternatives in the archive. It is also possible that some alternatives in the archive are not yet explored by the local search.

Obviously, the depot decision has long-term influences on the objectives. Exemplarily the comparison of the approximations between the 'original depot' and 'depot with identification number (ID) 39' are presented in Fig. 1. Note that the number 39 corresponds to identification number of the customer in the data set which has the x-coordinate 59 and y-coordinate 15. The whole approximation of the Pareto front for the 'original depot' lies below the other approximation, i.e. the approximation is considerable better. This presentation gives an idea how much influence a long-term decision, such as a 'bad' choice of the depot location, has on the inventory management and the vehicle routing. The approximations are compared after 235,000 number of evaluations, and e.g. not 300,000 evaluations as illustrated in Fig. 2 respectively Fig. 3, since all alternatives in the archive for 'depot with identification number 39' are already investigated by the local search.

The results presented in Fig. 2 for the 'original depot' and 'depot with identification number 12' (x-coordinate 31 and y-coordinate 32) are not as clear as the aforementioned findings. The gap between these approximations is much smaller compared to the approximations in Fig. 1. The advantage of choosing 'depot with ID 12' over the 'original depot' depends on the preferred part of the Pareto front. In

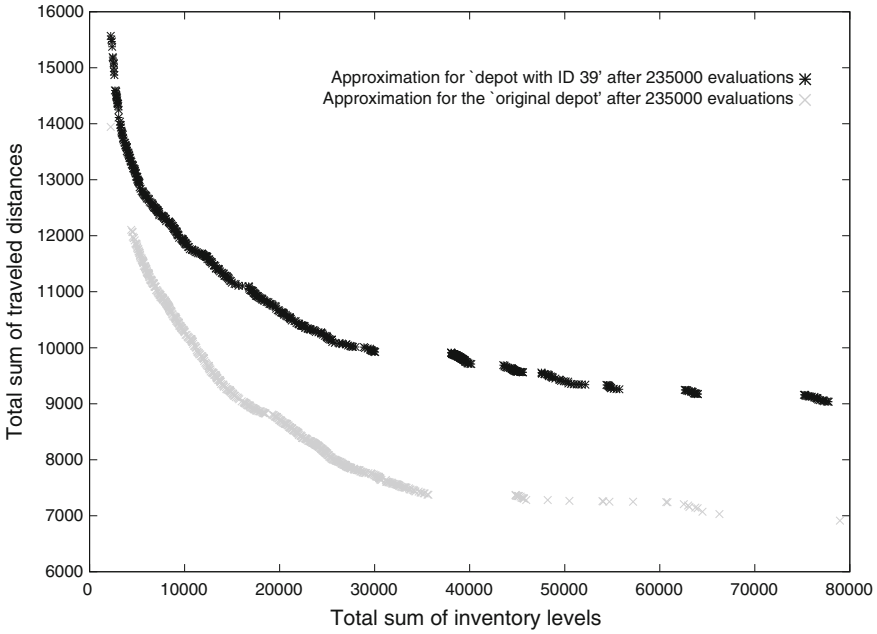


Fig. 1 Comparison of the approximations for the 'original depot' and 'depot with ID 39'

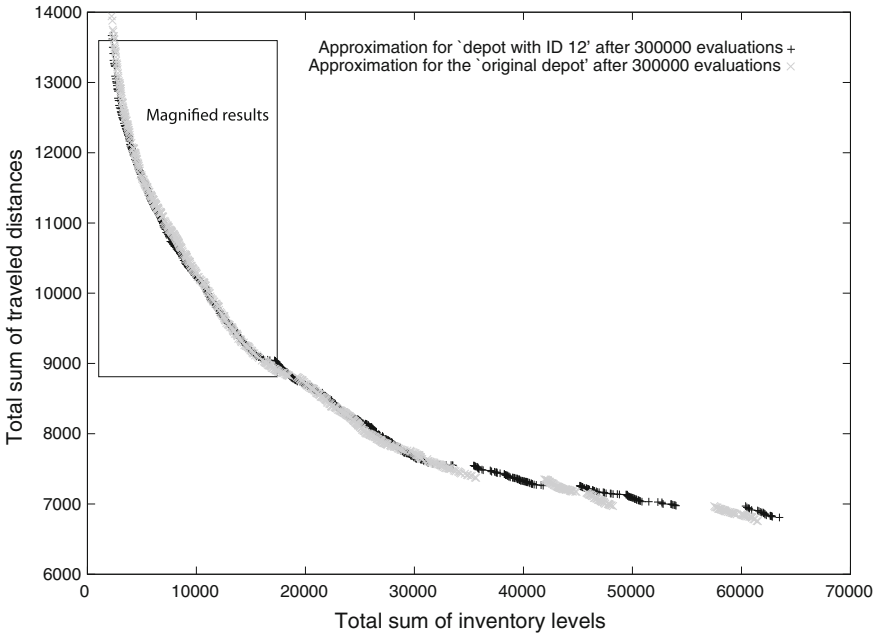


Fig. 2 Comparison of the approximations for the 'original depot' and 'depot with ID 12'

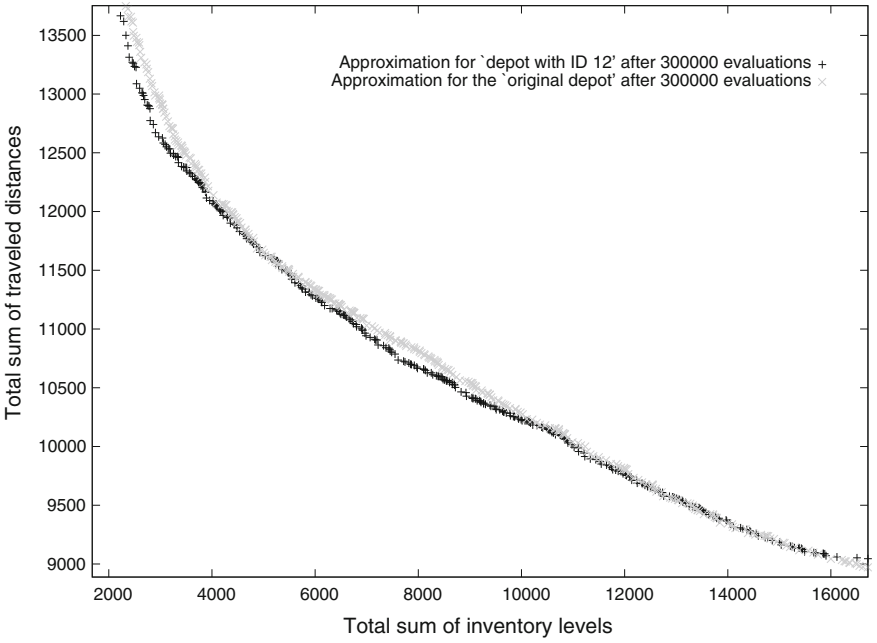


Fig. 3 Magnified results for the 'original depot' and 'depot with ID 12'

Fig. 3 magnified results are illustrated for the approximations at the extreme end of the Pareto front. Here, it is highlighted that the solution quality is mainly better when 'depot with ID 12' is selected. In order to identify the magnitude of the improvement, we compute the difference between the routing distances for any given inventory level. Over the entire approximation of the Pareto front, the best improvement for the 'depot selection with ID 12' is 3.31%. This reduction can be meaningful since such decisions are made for a long-term planning horizon.

4 Conclusion

We have conducted experiments to study the two proposed research questions. Our main contributions are twofold. The maximal influence on the inventory levels and the routing distances of a 'bad' depot decision is rather significant. We could show with straight-forward experiments for benchmark data sets with a given depot that savings in transportation costs (around 3%) can be achieved with a revised depot selection. Based on these findings, it seems promising to include depot decisions in the IRP for long-term planning decisions. However, counterexamples exist for *GS-b-01*, where the 'original depot' achieves better results. For future research this must be further analyzed for more test instances with different demand patterns.

References

1. Barthélemy, T., Geiger, M.J., Sevaux, M.: An extended solution representation for the biobjective inventory routing problem. In: Renatus, F., Kunze, R., Karschin, I., Geldermann, J., Fichtner, W. (eds.) Entscheidungsunterstützung durch Operations Research im Energie- und Umweltbereich, pp. 7–20. Shaker Verlag (2012)
2. Bertazzi, L., Savelsbergh, M., Speranza, M.G.: Inventory routing. In: Golden, B., Raghavan, S., Wasil, E. (eds.) The Vehicle Routing Problem—Latest Advances and New Challenges, pp. 49–72. Springer (2008)
3. Bertazzi, L., Speranza, M.G.: Inventory routing problems with multiple customers. EUR. J. Transp. Logist. **2**(3), 255–275 (2013)
4. Coelho, L.C.: Flexibility and Consistency in Inventory-Routing. Ph.D., HEC Montréal—Affiliée à l' Université de Montréal (2012)
5. Geiger, M.J., Sevaux, M.: The biobjective inventory routing problem—problem solution and decision support. In: Pahl, J., Reiners, T., Voß, S. (eds.) Network Optimization. Lecture Notes in Computer Science, vol. 6701, pp. 365–378. Springer, Berlin, Heidelberg (2011)
6. Guerrero, W.J., Prodhon, C., Velasco, N., Amaya, C.A.: A relax-and-price heuristic for the inventory-location-routing problem. Int. J. Adv. Manuf. Technol. **22**(1), 129–148 (2015)
7. Hiassat, A.H., Diabat, A.: A location-inventory-routing problem with perishable products. In: Proceedings of the 41st International Conference on Computers and Industrial Engineering (2011)
8. Huber, S., Geiger, M.J., Sevaux, M.: Simulation of preference information in an interactive reference point-based method for the bi-objective inventory routing problem. J. Multi-Criteria Decis. Anal. **22**(1–2), 17–35 (2015)

9. Liu, S., Lin, C.: A heuristic method for the combined location routing and inventory problem. Int. J. Adv. Manuf. Technol. **26**(4), 372–381 (2005)
10. Sevaux, M., Geiger, M.J.: Inventory routing and on-line inventory routing file format. Technical Report RR-11-01-01, Helmut-Schmidt-University (2011)
11. Wang, C., Ma, Z., Li, H.: Stochastic dynamic location-routing-inventory problem in closed-loop logistics system for reusing end-of-use products. Intell. Comput. Technol. Autom. (ICICTA) **2**, 691–695 (2008)

The Generalized Steiner Cable-Trench Problem with Application to Error Correction in Vascular Image Analysis

Eric Landquist, Francis J. Vasko, Gregory Kresge, Adam Tal, Yifeng Jiang and Xenophon Papademetris

Abstract The Cable-Trench Problem (CTP) is the problem of minimizing the cost to connect buildings on a campus to a central server so that each building is connected directly to the server via a dedicated underground cable. The CTP is modeled by a weighted graph in which the vertices represent buildings and the edges represent the possible routes for digging trenches and laying cables between two buildings. In this paper, we define the Generalized Steiner CTP (GSCTP), which considers the situation in which a subset of the buildings is connected to the server and also the possibility that trench costs vary because of vegetation or physical obstacles, for example. The GSCTP has several natural applications, but we will focus on its nontrivial and novel application to the problem of digitally connecting microCT scan data of a vascular network with fully automated error correction. The CTP and its variants are NP-hard. However, we show that modifications to Prim's algorithm find nearly optimal solutions to the GSCTP efficiently.

E. Landquist (✉) · F.J. Vasko
Department of Mathematics, Kutztown University,
Kutztown, PA 19530, USA
e-mail: elandqui@kutztown.edu

F.J. Vasko
e-mail: vasko@kutztown.edu

G. Kresge · A. Tal
Department of Computer Science and Information Technology,
Kutztown University, Kutztown, PA 19530, USA
e-mail: gkres121@live.kutztown.edu

A. Tal
e-mail: atal822@live.kutztown.edu

Y. Jiang · X. Papademetris
School of Medicine, Yale University, 310 Cedar Street,
208042, New Haven, CT 06520-8042, USA
e-mail: jiang1feng@gmail.com

X. Papademetris
e-mail: xenophon.papademetris@yale.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_52

1 Introduction

The Cable-Trench Problem (CTP) was first described in [6] and establishes a continuum between the Minimum Spanning Tree and Shortest Path Tree Problems. The name "Cable-Trench" comes from the problem of minimizing the cost to connect buildings on a campus to a central server so that each building is connected directly to the server via a dedicated underground cable. The problem is modeled as a weighted graph in which the buildings are represented by vertices and the edges represent the possible routes for digging trenches and laying cables between two buildings. Weights on the edges generally represent distance. The Generalized CTP (GCTP) considers the possibility that the cost of digging a trench varies because of vegetation, soil composition, or physical obstacles, for example [7]. The Generalized Steiner CTP (GSCTP) further supposes that some subset of the buildings is connected to the server, though cables may be routed through any building.

In Sect. 2, we describe the application of the GSCTP to the problem of digitally reconstructing a blood vessel network from microCT scan image data and eliminating errors in the data. We describe heuristics that we used to quickly compute nearly optimal solutions to the GSCTP in Sect. 3 and tabulate results of our experiments in Sect. 4. The paper closes with some conclusions and areas of future work in Sect. 5. Here, we give a graph-theoretic description of the GSCTP.

Let $G = (V, E)$ be a connected graph with vertex set $V = \{v_1, \ldots, v_n\}$, root vertex v_1, edge set E, and $s_{ij} \geq 0$ and $t_{ij} \geq 0$ the "cable" and "trench" weights of the edge $(v_i, v_j) \in E$, respectively. Let $F \subseteq V$ be the set of terminal vertices, $N \subseteq V$ the set of nonterminal vertices, and let γ and τ denote the per-unit cable and trench costs, respectively. We define the GSCTP as the problem of finding a tree $T = (V_T, E_T)$, such that $\{v_1\} \cup F \subseteq V_T \subseteq V$ and $E_T \subseteq E$, which minimizes $\gamma w_c(T) + \tau w_t(T)$, where

$$w_c(T) = \sum_{v_k \in F} \sum_{(v_i, v_j) \in \mathscr{P}(v_1, v_k)} s_{ij} \quad \text{and} \quad w_t(T) = \sum_{(v_i, v_j) \in E_T} t_{ij} \tag{1}$$

are the total cable weight of T and total trench weight in T, respectively, and $\mathscr{P}(v_1, v_k) \subseteq E_T$ is the path in T from v_1 to the terminal vertex v_k. Vertices in $V_T \cap N$ are called Steiner vertices. The CTP is the special case in which $N = \emptyset$ and $s_{ij} = t_{ij}$ for all i and j. Further, if $\gamma > 0$ and $\tau = 0$, then a solution to the CTP is any shortest path spanning tree of G with root vertex v_1. In contrast, if $\tau > 0$ and $\gamma = 0$, then a solution to the CTP is any minimum spanning tree of G. Note that if $\tau = 0$, then an optimal solution could contain cycles formed from "empty" trenches. We will not consider such solutions because in practice, we want to utilize every trench.

We refer the reader to [7] for a description of further applications and extensions of the CTP. To motivate our definition of the GSCTP, however, we will describe a nontrivial application to vascular image analysis due to Jiang et al. [2].

2 Application to Vascular Image Analysis

A massive set of discrete points, representing the locations of blood vessels, are first detected from 3D medical images, such as CT and microCT. The vessel radii at these points can also be estimated from the images. These points correspond to the vertices, V, of a complete graph $G = (V, E)$, with edge weights determined by Euclidean length, vessel segment volume, or some physiological factor. The goal is to digitally represent the vessel network (vasculature) as a subtree $T \subseteq G$ as accurately as possible in order to assist critical vasculature-related research, including angiogenesis and cancer detection. This task currently constitutes a bottleneck in quantitative vascular research [8]. Prior to the GCTP model of [2, 7], the best methods computed the minimum spanning tree of G, but the results depended heavily on manual correction [1, 3, 5]. Specifically, [2, 7] applied Murray's Minimum Work Principle [4], which states that any vascular network tends to minimize the total work due to blood flow resistance and metabolic support for the blood volume. These two factors are proportional to the cable cost and trench cost in the GCTP, respectively. Thus, the vessel connection problem is formulated as a GCTP with cable weights $s_{ij} = \ell(e_k)/r(e_k)$ and trench weights $t_{ij} = \ell(e_k)r(e_k)^2$, where $\ell(e_k)$ is the length of the blood vessel represented by the edge $e_k = (v_i, v_j)$ and $r(e_k)$ is the radius of the vessel at v_j. In this application, if $\gamma = 1$, then $35000 \leq \tau \leq 175000$ is an appropriate range. We refer the reader to [2] for the technical details of their derivation and to [7] for results on the GCTP treatment of this problem.

In real image analysis scenarios, however, the data invariably contains errors, i.e., false positive vessel points detected from images. One can determine the leaves of the solution tree, typically those points within the region perfused by the vascular tree. We can therefore model the vascular imaging problem as a GSCTP by letting F be the set of known leaf vertices. The set of errors, then, is a subset of N. We assume that the optimal solution to a GSCTP model of a vessel connection problem will yield an image as close as possible to the actual vascular network.

Vasko et al. showed that the CTP is NP-hard [6], so the GSCTP is NP-hard. Thus, it is computationally infeasible to determine optimal solutions of very large GSCTPs, such as those arising from the application at hand. In order to efficiently find nearly optimal solutions of the GSCTP, we modified Prim's algorithm.

3 Modifications to Prim's Algorithm

In [7], Vasko et al. extended Prim's algorithm to find nearly optimal solutions of the GCTP. In this section, we describe further modifications that allow one to find nearly optimal solutions to the GSCTP. First, we describe a modification of Prim's algorithm, which we call the Generalized Steiner Modified Prim's heuristic (`GSMPrim`). This will include what we call a *benefit* function, which is designed to encourage the selection of terminal vertices as well as Steiner vertices that are adjacent to or

sufficiently close to multiple terminal vertices. We then describe two variants of GSMPrim: one semi-greedy and deterministic and one partially stochastic.

In order to define the benefit function, we first let $|\cdot| : E \to \mathbb{R}_{\geq 0}$ be some fixed edge metric (e.g., cable or trench weight), and set $B \in \mathbb{R}_{\geq 0} \cup \{\infty\}$ so that $|(v,w)| < B$ implies that $v, w \in V$ are near each other. The metric $|\cdot|$ and bound B will vary depending on the application. Now, the benefit function, $b : V \to \mathbb{N}_0$, is defined

$$b(v) = \begin{cases} 2 + \#\{w \in F : (v,w) \in E \setminus E_T \text{ and } |(v,w)| < B\} & \text{if } v \in F \\ \#\{w \in F : (v,w) \in E \setminus E_T \text{ and } |(v,w)| < B\} & \text{if } v \in N. \end{cases} \tag{2}$$

We also define a positive multiplier M and let $W = (\tau/\gamma)M \max_{e \in E}\{|e|\}$.

Algorithm 1: Generalized Steiner Modified Prim's Heuristic (GSMPrim)

Input : $G = (V, E)$, F, s_{ij} and t_{ij}, γ, τ, B, and W
Output: A tree $T = (V_T, E_T) \subseteq G$, such that $\{v_1\} \cup F \subseteq V_T$

1 $V_T := \{v_1\}$, $E_T := \{\}$, $cost := d := \{\infty, \infty, \ldots, \infty\}$, $Pre := \{1, 1, \ldots, 1\}$;
2 **for** $2 \leq i \leq n$ **do**
3 $\quad cost[i] := \gamma s_{1i} + \tau t_{1i}$
4 **while** $\{v_1\} \cup F \nsubseteq V_T$ **do**
5 $\quad m := \text{index}\left(\min\left\{cost[i] - Wb(v_i) : v_i \in V \setminus V_T\right\}\right)$;
6 $\quad V_T := V_T \cup \{v_m\}$ and $E_T := E_T \cup \{\left(v_{Pre[m]}, v_m\right)\}$;
7 $\quad$ **for** $v_i \in V \setminus V_T$ *such that* $(v_m, v_i) \in E$ **do**
8 $\quad\quad$ **if** $cost[i] > \gamma(d[m] + s_{mi}) + \tau t_{mi}$ **then**
9 $\quad\quad\quad d[i] := d[m] + s_{mi}$, $cost[i] := \gamma(d[m] + s_{mi}) + \tau t_{mi}$, $Pre[i] := m$;
10 Remove all leaves in N from V_T so that all leaves of $T = (V_T, E_T)$ are in F.

Since we add a new vertex and edge with each iteration of the while loop, GSMPrim will terminate. Moreover, it requires $O\left(|V|^2\right)$ time and space.

The two variants of GSMPrim are multi-pass modifications that generate multiple solution trees. The semi-greedy variant, SG-GSMPrim, selects the kth best edge and vertex at Step 5 of GSMPrim for the first edge of the kth solution tree T. Thereafter, it selects every edge and vertex in a greedy fashion. The partially stochastic variant, PS-GSMPrim, selects an initial fraction of the vertices and edges at Step 5 in a stochastic manner, and proceeds in a greedy fashion thereafter. Specifically, we let $\left[p_1, p_2, \ldots, p_r\right]$ be a probability distribution, with $p_i \geq p_{i+1}$ for all $1 \leq i < r$. At Step 5, the ith best vertex and edge is selected with probability p_i, for $1 \leq i \leq r$. In this way, PS-GSMPrim can generate any number of solution trees.

Note that in the application to vascular image analysis error correction, we assume that nonterminal leaves are in fact errors in the imaging process. Thus, Step 10 of GSMPrim and its variants automatically eliminates errors from the image data.

4 Results

We tested GSMPrim and its variants on a pair of small graphs and on a 25001-vertex data set generated from a microCT scan of the vasculature of a mouse leg. In each case, we let $\gamma = 1$ and considered a collection of values of τ. For the benefit function, we used $|e| = \ell(e)$ and $B = (1/10)\max_{e\in E}\{\ell(e)\}$.

For Tables 1 and 2, we used five sets of terminal vertices on each graph and averaged the results. In Table 1, we compared GSMPrim with $M = 0$ and $M = 1$ to the optimal solutions, which were found using LINDO. The last column shows the percentage improvement in GSMPrim when using the benefit function. In Tables 1 and 2, the optimal solutions are used as the benchmark to test our heuristic.

The graph for Table 2 is taken from the microCT scan data: the root and its closest 20 points. For Tables 2 and 3, we tested multipliers M from the set $\{10^{-5}, 10^{-4}, 10^{-3}, 0.01, 0.1, 0.2, \ldots, 1.5, 2, 2.5, 3\}$. The last three columns in Tables 2 and 3 give the percentage improvement of GSMPrim with the benefit function, SG-GSMPrim, and PS-GSMPrim over GSMPrim with $M = 0$, respectively. For each example, we ran 20 iterations of SG-GSMPrim and 30 of PS-GSMPrim using the probability distribution $[1/3, 2/9, 2/9, 1/9, 1/9]$. The first 15 iterations chose the first five vertices stochastically and the next 15 did so with the first ten vertices.

Table 1 GSMPrim on a 9-vertex SCTP from Example 4 of [6]

τ	Opt.	$M = 0$	% Dev.	$M = 1$	% Dev.	% Impr.
0.1	55.54	55.54	0	55.54	0	0
1	84	85.80	2.14	84	0	2.10
5	201.6	249.4	23.7	204.2	1.29	18.1
30	922.6	1134.4	22.5	929.2	0.72	18.1

Table 2 GSMPrim on a 21-vertex GSCTP

τ	$M = 0$, % Dev./Opt.	Best M	% Impr./$M = 0$	SG % Impr.	PS % Impr.
0.01	4.19	0.00001	0.0003	0	0
1	4.19	0	0	0.00009	0
5	4.19	0	0	0.006	0
10	4.19	0	0	0.006	0
100	4.06	0	0	0.006	0
10000	5.41	0.001	6.51	8.43	3.58
50000	16.2	0.001	4.39	3.18	6.33
100000	16.2	0.001	1.76	2.54	6.75
150000	15.2	0.001	3.53	2.34	6.80

Table 3 GSMPrim on a 25001-vertex Vascular Data Set

τ	Best M	% Impr./$M = 0$	SG % Impr.	PS % Impr.
10000	0.01	4.59	1.31	0.63
50000	0.01	4.35	0.93	1.30
100000	0.01	2.15	0.90	1.42
150000	0.01	0	0.85	1.77

In Table 3, we focus on the vascular image analysis problem and data, so we only tested those values of τ appropriate for this application. We chose 22500 vertices at random to be the set F. In this case, we ran 30 iterations each of SG-GSMPrim and PS-GSMPrim. The first 15 iterations of PS-GSMPrim chose the first 1500 vertices stochastically and the last 15 iterations chose the first 2500 vertices stochastically. GSMPrim with $M = 0$ is used as a benchmark heuristic.

We ran GSMPrim and its variants in MATLAB® on a PC running Windows 7 Professional with an Intel I7-3930K 3.2 GHz processor and 16 GB of RAM. Each run on the 25001-vertex examples took an average of 34.0 s.

5 Conclusions and Future Work

In this paper, we defined the GSCTP and applied it to solve the problem of algorithmic error correction in vascular image analysis. We developed three efficient variants of a heuristic based on Prim's algorithm that are capable of finding nearly optimal solutions to GSCTPs. Generally, as τ increased, the accuracy of GSMPrim declined, but its variants yielded significant improvements. In particular, using the benefit function was the most effective approach for the largest graphs.

Since the CTP and its variants are a relatively new and unexplored area of study, there are several avenues to consider for future work, both theoretically and experimentally. For the GSCTP in particular, we would like to experiment with genetic algorithms and use LP relaxation techniques to determine good lower bounds of optimal solutions, to test the effectiveness of GSMPrim on larger graphs.

Acknowledgements The authors thank Dr. Albert Sinusas and Zhenwu Zhuang, Yale University School of Medicine, for providing microCT scan image data of a mouse leg for our experiments.

References

1. Bullitt, E., Aylward, S., Liu, A., Stone, J., Mukherji, S., Coffey, C., Gerig, G., Pizer, S.: 3D graph description of the intracerebral vasculature from segmented MRA and tests of accuracy by comparison with X-ray angiograms. In: Kuba, A., Sámal, M., Todd-Pokropek, A. (eds.) Proceedings of IPMI. LNCS, vol. 1613, pp. 308–321. Springer, Berlin (1999)

2. Jiang, Y., Zhuang, Z., Sinusas, A., Staib, L., Papademetris, X.: Vessel connectivity using Murray's hypothesis. In: Fichtinger, G., Martel, A., Peters, T. (eds.) Proceedings of MICCAI 2011. LNCS, vol. 6893, pp. 528–536. Springer, Berlin (2011)
3. Jomier, J., Ledigarcher, V., Aylward, S.: Automatic vascular tree formation using the Mahalanobis distance. In: Duncan, J., Gerig, G. (eds.) Proceedings of MICCAI 2005. LNCS, vol. 3750, pp. 806–812. Springer, Berlin (2005)
4. Murray, C.: The physiological principle of minimum work, I. the vascular system and the cost of blood volume. Proc. Natl. Acad. Sci. **12**(3), 207–214 (1926)
5. Szymczak, A., Stillman, A., Tannenbaum, A., Mischaikow, K.: Coronary vessel trees from 3D imagery: a topological approach. Med. Image Anal. **10**(4), 548–559 (2006)
6. Vasko, F., Barbieri, R., Rieksts, B., Reitmeyer, K., Stott, K.: The cable trench problem: combining the shortest path and minimum spanning tree problems. Comput. Oper. Res. **29**(5), 441–458 (2002)
7. Vasko, F., Landquist, E., Kresge, G., Tal, A., Jiang, Y., Papademetris, X.: A simple and efficient strategy for solving very large-scale generalized cable-trench problems. Netw. Int. J. **67**(3), 199–208 (2016)
8. Zagorchev, L., Oses, P., Zhuang, Z., Moodie, K., Mulligan-Kehoe, M., Simons, M., Couffinhalt, T.: Micro computed tomography for vascular exploration. J. Angiogenesis Res. **2**(1), 7 (2010)

Ensemble Techniques for Scheduling in Heterogeneous Wireless Communications Networks

David Lynch, Michael Fenton, Stepan Kucera, Holger Claussen and Michael O'Neill

Abstract Operators deploy Small Cells in high traffic regions to boost the capacity of their wireless networks. However, User Equipments (UEs) at Small Cell edges experience severe interference from neighbouring high-powered Macro Cells. A fair trade-off between cell-edge and cell-centre performance can be realised by intelligently scheduling Small Cell attached UEs. Grammar-based Genetic Programming is employed to learn models that map measurement reports to schedules on a millisecond timescale. The evolved schedulers are then aggregated into ensembles. The proposed system significantly outperforms a state of the art benchmark algorithm and is within 10% of the estimated optimum.

1 Introduction

Traditional single-tiered cellular networks are struggling to cope with exponentially rising demand. Capacity can be increased by supplementing the existing Macro Cell (MC) tier with Small Cells (SCs). These lower-powered base stations provide a local capacity boost in traffic hotspots. The resulting two-tiered configuration is known as a Heterogeneous Network or 'HetNet'.

Operators such as AT&T Inc. are aggressively densifying with SCs because both cell tiers can reuse the same scarce and expensive bandwidth. Unfortunately, severe interference arises at SC edges in channel sharing HetNets. Cell-edge conditions can be improved by forcing MCs to mute during so-called 'Almost Blank Subframes' (ABSFs) [2]. Note that 'subframes' are 1 ms intervals during which cells transmit packets to their attached User Equipments ('UEs' refer to smartphones, tablets etc.).

It is typically suboptimal to schedule all users attached to a SC s (i.e. the set $\mathcal{A}_s$) in every subframe. Better fairness is achieved by intelligently scheduling different

D. Lynch (✉) · M. Fenton · M. O'Neill
Natural Computing Research and Applications Group, School of Business, University College Dublin, Dublin, Ireland
e-mail: david.lynch.1@ucdconnect.ie

S. Kucera · H. Claussen
Bell Laboratories, Nokia, Dublin, Ireland

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_53

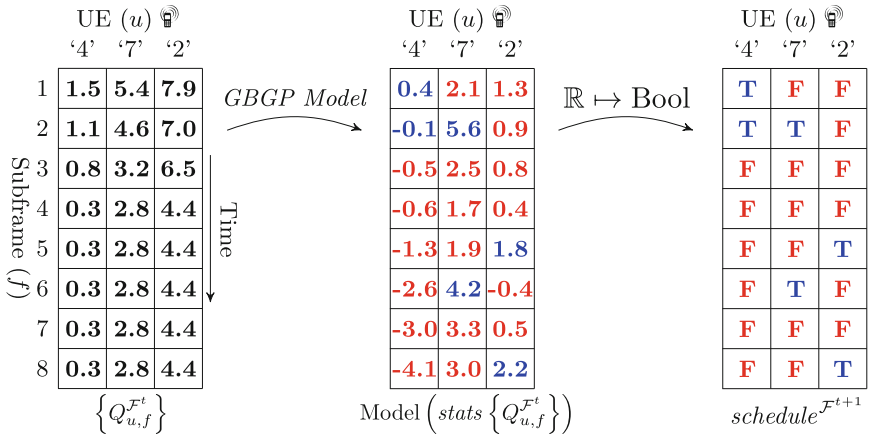


Fig. 1 Mapping measurement reports to schedules

subsets of $\mathcal{A}_s$ to receive packets in successive subframes. Fairness is vital because all customers must experience an acceptable quality of service. This paper proposes a framework for automatically constructing near optimal schedulers that execute on the required millisecond timescale.

2 Problem Definition and Previous Work

Let $Q^{\mathcal{F}^t}_{u,f} := \log_2\left(1 + SINR^{\mathcal{F}^t}_{u,f}\right)$ denote the channel quality experienced by UE u in subframe f of frame $\mathcal{F}^t$, where $SINR^{\mathcal{F}^t}_{u,f}$ is the signal that u receives from its serving cell in f divided by interference and noise. UEs report measurements of $Q^{\mathcal{F}^t}_{u,f}$ to their serving cell after every frame spanning 8 subframes[1] or 8 (ms). Shannon's formula gives the rate at which information flows through the wireless channel to UE u, in subframe f of frame $\mathcal{F}^t$:

$$R^{\mathcal{F}^t}_{u,f} = \frac{B}{N^{\mathcal{F}^t}_f} \times \log_2\left(1 + SINR^{\mathcal{F}^t}_{u,f}\right), \tag{1}$$

where, $R^{\mathcal{F}^t}_{u,f}$ is the downlink rate, $B = 20$ MHz is the fixed bandwidth, and $N^{\mathcal{F}^t}_f$ is the number of UEs receiving data from u's serving cell in f.

The leftmost panel of Fig. 1 displays typical values of $Q^{\mathcal{F}^t}_{u,f}$ over frame $\mathcal{F}^t$, for a SC (s) with three attached UEs (let $\mathcal{A}^{\mathcal{F}^t}_s$ denote the set of UEs attached to s in frame $\mathcal{F}^t$). UEs and subframes are represented by columns and rows respectively. GBGP

[1] Canonically $|\mathcal{F}^t| = 40$, but WLOG schedules are computed for $f = \{1 \ldots 8\}$, c.f. [3].

is employed to learn a mapping from statistics over the set $\left\{ Q_{u,f}^{\mathcal{F}^t} | u \in \mathcal{A}_s^{\mathcal{F}^t}, f \in \mathcal{F}^t \right\}$, to the schedule for s. The real-valued outputs of the model (central panel) are interpreted as a Boolean schedule (rightmost panel), which s will observe in frame $\mathcal{F}^{t+1}$. Each UE is forced to receive data in exactly two subframes by setting the largest two cells in each column from the central panel to '*True*' and the remaining cells to '*False*' [3]. For instance, UE 4 will receive packets from s in subframes $f = \{1, 2\}$ but not in $f = \{3 \dots 8\}$.

The quality or "fitness" of the schedule in Fig. 1 is given by the sum-log-rates (SLR) metric of fairness:

$$SLR_s := \sum_{u \in \mathcal{A}_s^{\mathcal{F}^{t+1}}} \log_e \left(\frac{1}{8} \sum_{f=1}^{8} R_{u,f}^{\mathcal{F}^{t+1}} \right). \tag{2}$$

Equation 1 implies that $R_{u,f}^{\mathcal{F}^t} \propto Q_{u,f}^{\mathcal{F}^t} / N_f^{\mathcal{F}^t}$. Therefore, knowledge of the reported channel qualities and the schedule are sufficient to evaluate Eq. 2. For example, $SLR_s = 48.72$ where s is the SC depicted in Fig. 1 (assuming for simplicity that $Q_{u,f}^{\mathcal{F}^{t+1}} = Q_{u,f}^{\mathcal{F}^t}$ and $\mathcal{A}_s^{\mathcal{F}^{t+1}} = \mathcal{A}_s^{\mathcal{F}^t}$). A scheduler is optimal if it maximises SLR_s.

A state of the art algorithm for scheduling in HetNets was proposed by López-Pérez and Claussen [1]. Two queues are initialised for a given SC: $\mathcal{Q}_{non-ABSF}$ and $\mathcal{Q}_{ABSF}$. UEs are transferred between queues, subject to constraints, in order to equalise the downlink rates of the two worst performers in each. When the algorithm converges, the SC schedules $u \in \mathcal{Q}_{non-ABSF}$ in non-ABSFs and $u \in \mathcal{Q}_{ABSF}$ in ABSFs. The scheduler from [1] serves as a benchmark for the method proposed in this paper.

Previous work by the authors [3] has demonstrated the suitability of Grammar-based Genetic Programming (GBGP) [4] as a framework for automatically devising SC schedulers. The following sections outline an ensemble approach that leverages the stochastic properties of GBGP.

3 Experiments

A 3.61 km^2 area of downtown Dublin was simulated. Realistic channel quality reports were computed by modelling the distribution of buildings, open spaces and waterways. These training data were generated in a network containing 30 SCs and 21 MCs. The set $\left\{ Q_{u,f}^{\mathcal{F}^t} | u \in \mathcal{A}_s^{\mathcal{F}^t} \right\}$ for SC s represented a single training case. The training set was instantiated with three hundred cases from ten different frames in order to encourage good generalisation. Cell powers and MC muting patterns were set according to the heuristics in [1]. The grammar from [3] was instrumented to evolve functional expressions (schedulers) using GBGP. The same evolutionary parameters were adopted from [3] except $\#\,gens := 200$. Finally, the evolved models from 1500 independent runs were arbitrarily grouped into 30 ensembles.

An individual model's fitness on a training case s was given by,

$$RF_{s,model} := \left(\frac{SLR_{s,model} - SLR_{s,baseline}}{SLR_{s,CMA} - SLR_{s,baseline}} \right) \times 100\%, \tag{3}$$

where, $RF_{s,model}$ expresses the model-generated schedule's fitness relative to the performance of Covariance Matrix Adaptation-Evolutionary Strategy (CMA) and a greedy baseline. CMA can be used to compute highly optimised schedules offline but it is far too slow for on the fly optimisation in real HetNets. The baseline greedily schedules all $u \in \mathcal{A}_s^{\mathcal{F}^t}$ in every subframe. Equation 3 evaluated to $\approx 100\%$ if a near optimal schedule was generated for case s and $\leq 0\%$ if the greedy baseline was not surpassed. A model's overall fitness was given by the average of $RF_{s,model}$ over all training cases. Mean end-of-run fitness was 69% for the 1500 individual models, but performance (on training data) was boosted to 92% by allowing the models to cooperate as ensembles.

3.1 Performance on Test Data

Schedules for frame $\mathcal{F}^{t+1}$ must be computed in frame $\mathcal{F}^t$ based on $\left\{ Q_{u,f}^{\mathcal{F}^t} \right\}$. The interval between $\mathcal{F}^t$ and $\mathcal{F}^{t+1}$ is sufficient to generate hypothesis schedules from an ensemble of 50 models. In real time, Eq. 2 is evaluated for each hypothesis against $\left\{ Q_{u,f}^{\mathcal{F}^t} | u \in \mathcal{A}_s^{\mathcal{F}^t} \right\}$ and the best schedule (w.r.t. SLRs) is used by s in $\mathcal{F}^{t+1}$. The proposed ensemble method exploits the stochastic nature of GBGP since evolved models tend to be semantically unique and hence they admit non-overlapping errors. In the following analysis, UEs are displaced slightly between $\mathcal{F}^t$ and $\mathcal{F}^{t+1}$ to simulate the mobility of customers between frames. Thus, schedules are based on slightly outdated reports.

Table 1 compares the benchmark and evolutionary methods on unseen test data. Equation 3 is averaged for all 30 SCs (test cases) in the HetNet over 100 unseen frames. A one-way ANOVA reveals that there is a significant ($p = 0.000$) difference between the methods across $n = 100$ frames (with $F(3, 396) = 6411.5$), and Tukey's post-hoc analysis confirms that the group means are mutually significantly different at $\alpha = 0.05$. Schedules computed by CMA in $\mathcal{F}^t$ are 1.2% from the estimated optimum in $\mathcal{F}^{t+1}$ due to UE mobility. The ensemble is within 8.1% of the estimated

Table 1 Average relative fitness of the methods over 100 test frames

	Benchmark	Best-of-Ensemble	Ensemble	CMA
RF^{avg} (%)	20.2 ± 8.2	84.0 ± 2.6	91.9 ± 1.4	98.8 ± 2.4

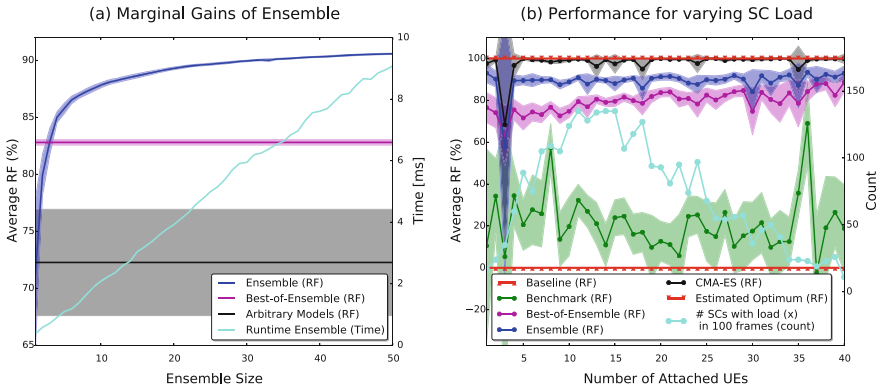


Fig. 2 Performance w.r.t. ensemble size (*left*) and cell load (*right*)

optimum which is impressive since it executes on the timescale of a frame. Highly fit 'best-of-ensemble' models are outperformed by the ensembles. Finally, the benchmark admits much lower fitness and it is less stable than the evolved models.

Figure 2a plots the average performance of the 30 ensembles on the test set against ensemble size. Only three models working cooperatively are needed to surpass the best individual model from 50 runs. The grey line describes thirty models that were selected at random. Its relatively wide 95% confidence interval suggests that multiple runs should be performed when building a scheduler using GBGP. Execution time increases linearly with ensemble size. Figure 2b reveals that the ensemble and best-of-ensemble models significantly outperform the benchmark for almost all cell loads (i.e. $|\mathcal{A}_s|$). The non-negligible optimality gap w.r.t. CMA illustrates an opportunity for further gains in future work.

3.2 Semantics

Figure 3 visualises the semantics. Deep red in cell (u, f) indicates that u is scheduled in f often, and deep blue implies that u is rarely scheduled in f. The benchmark unschedules cell-centre UEs in protected subframes (i.e. ABSFs), therein liberating bandwidth for cell-edge UEs. Conversely, cell-centre UEs receive most of the bandwidth during less protected subframes (since cell-edge UEs are unscheduled). The central heatmap illustrates how evolved models emulate the benchmark's core strategy. However, the search heuristic is stochastic so different models generate distinct hypotheses. Thus, an ensemble can produce highly specialised schedules (rightmost heatmap).

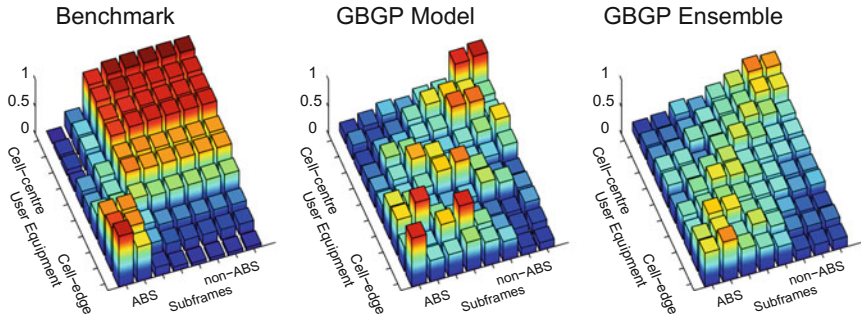


Fig. 3 Visualising the semantics

4 Conclusions and Future Work

Multiple models can be executed in the dead time between frame $\mathcal{F}^t$ and $\mathcal{F}^{t+1}$, thus yielding several hypothesis schedules for a SC, of which the best is implemented during $\mathcal{F}^{t+1}$. Independent runs of GBGP yield semantically unique solutions, so that the hypotheses are well dispersed in the solution space. Thus, the ensemble members tend to make non-overlapping errors. The proposed method approximates the estimated optimum given by running CMA offline. Crucially, all models can be executed on the timescale of a single frame. Future work could co-evolve the ensemble members so that they cooperate more effectively.

Acknowledgements This research is based upon works supported by the Science Foundation Ireland under grant 13/IA/1850.

References

1. López-Pérez, D., Claussen, H.: Duty cycles and load balancing in HetNets with eICIC almost blank subframes. In: 2013 IEEE 24th International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC Workshops), pp. 173–178. IEEE (2013)
2. Lopez-Perez, D., Guvenc, I., De la Roche, G., Kountouris, M., Quek, T.Q., Zhang, J.: Enhanced intercell interference coordination challenges in heterogeneous networks. IEEE Wirel. Commun. **18**(3), 22–30 (2011)
3. Lynch, D., Fenton, M., Kucera, S., Claussen, H., O'Neill, M.: Evolutionary learning of scheduling heuristics for heterogeneous wireless communications networks. In: Proceedings of the 2016 on Genetic and Evolutionary Computation Conference, pp. 949–956. ACM (2016)
4. Mckay, R.I., Hoai, N.X., Whigham, P.A., Shan, Y., O'Neill, M.: Grammar-based genetic programming: a survey. Genet. Program. Evolvable Mach. **11**(3–4), 365–396 (2010)

A Heuristic for Solving the Maximum Dispersion Problem

Mahdi Moeini and Oliver Wendt

Abstract In this paper, we investigate solving the Maximum Dispersion Problem (MaxDP). For a given set of weighted objects, the MaxDP consists in partitioning this set into a predefined number of groups, such that the overall dispersion of elements, assigned to the same group, is maximized. Furthermore, each group has a target weight and the total weight of each group must be within a specific interval around the target weight. It has been proven that the MaxDP is NP-hard and, consequently, difficult to solve by classical exact methods. In this work, we use variants of Variable Neighborhood Search (VNS) for solving the MaxDP. In order to evaluate the efficiency of VNS, we carried out some numerical experiments on randomly generated instances. The results of the VNS is compared with the solutions provided by the solver Gurobi. According to our results, the VNS gives high quality solutions for small instances and, in solving large instances, it provides some decent solutions for all instances; however, Gurobi fails to provide any solution for some of them.

1 Introduction

An important class of optimization problems concerns the partitioning of a set of objects into disjoint groups with the objective of minimizing or maximizing a predefined objective function [1, 3, 6, 8, 10, 11]. Sometimes, the objective is minimizing a function that defines the distance or dissimilarity between objects. However, there are also applications, where we want to maximize the dissimilarity (see, e.g., [5, 6, 8], and references therein). In this context, we focus on the construction of dispersed groups. An example for the construction of dispersed groups arises in constructing learning groups of students [5]. One of such combinatorial problems is the

M. Moeini (✉) · O. Wendt
Chair of Business Information Systems and Operations Research (BISOR),
Technical University of Kaiserslautern, Postfach 3049, Erwin-Schrödinger-Str., 67653 Kaiserslautern, Germany
e-mail: mahdi.moeini@wiwi.uni-kl.de

O. Wendt
e-mail: wendt@wiwi.uni-kl.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_54

Maximum Dispersion Problem (MaxDP). In this problem, we mainly partition a set of objects into a predefined number of groups such that the dispersion between objects assigned to the same group is maximized. This problem has been introduced by Fernández et al. [5]. The MaxDP is closely related to Maximum Diversity Problem that consists in selecting a maximally diverse subset of objects such that the diversity is measured by the sum of distances between chosen objects [6, 8].

It has been proven that the MaxDP is NP-hard and, consequently, difficult to solve by means of the classical exact methods [5]. Due to this fact, using a heuristic for solving the MaxDP is a natural choice. Variable Neighborhood Search (VNS) is an efficient heuristic that has been successfully used for solving numerous combinatorial optimization problems [2, 7]. Hence, the aim of this paper consists in evaluating a variant of the VNS heuristic for solving the MaxDP. More precisely, by taking into account the structure of the MaxDP, we design a VNS for solving the MaxDP. In order to evaluate the efficiency of the VNS for solving the MaxDP, we generated some random instances and solve them by means of the proposed VNS approach. The results of the VNS are compared with the solutions provided by Gurobi, that solves an Integer Programming formulation of the MaxDP. According to the preliminary numerical results, the VNS gives high quality solutions for small instances and, in solving large instances, it provides some solutions for all instances; however, Gurobi fails to provide any solution for some of them. In solving medium sized instances, Gurobi has a better performance.

The remainder of this paper is organized as follows. In Sect. 2, we describe the MaxDP problem and its mathematical programming models. Section 3 is devoted to the presentation of the proposed VNS heuristic. The results of our computational experiments are presented in Sect. 4. Some conclusions are drawn in Sect. 5.

2 The Maximum Dispersion Problem

In this section, we describe the MaxDP and present the mathematical formulations of the problem. Suppose that a set $V = \{1, \ldots, n\}$ of n objects is given and we want to distribute them into m disjoint sets $c \in C = \{1, \ldots, m\}$. Each object $i \in V$ has a weight of $a_i \geq 0$ and we assume that their sum is $A = \sum_{i \in V} a_i$. Furthermore, each group $c \in C$ has a target weight $M_c \geq 0$ and the sum of weights of the objects in any group needs to meet its target weight with respect to an permitted deviation $\alpha \geq 0$, i.e., the total weight of the group c must belong to $[(1-\alpha)M_c, (1+\alpha)M_c]$, where $c \in C$. We suppose that there is no object having a weight which allows it to make a singleton group on its own. Finally, the distance between objects i and j is denoted by d_{ij}.

In order to present the mathematical model of the MaxDP, we define, for each $i \in V, c \in C$, the decision binary variable x_{ic} that is 1 if object i is assigned to group c, otherwise $x_{ic} = 0$. Using these notations, we have the following nonlinear mathematical programming model for the MaxDP ($MaxDP_{NLP}$):

$$\text{Max Min}_{i,j\in V, c\in C} \quad \frac{d_{ij}}{x_{ic}x_{jc}} \tag{1}$$

$$\text{s.t.} \quad \sum_{c\in C} x_{ic} = 1, \qquad \forall i \in V, \tag{2}$$

$$\sum_{i\in V} a_i x_{ic} \geq (1-\alpha)M_c, \qquad \forall c \in C, \tag{3}$$

$$\sum_{i\in V} a_i x_{ic} \leq (1+\alpha)M_c, \qquad \forall c \in C, \tag{4}$$

$$x_{ic} \in \{0,1\}, \qquad \forall i \in V, c \in C. \tag{5}$$

In this model, the nonlinear objective function (1) maximizes the minimal distance between any pair of objects that belong to a same group. Due to the constraints (2), each object is assigned to only one group. The constraints (3) and (4) are the *balancing constraints* and force each group to respect the weight limits.

The *MaxDP* model also has linear programming formulations. One of them is motivated by the covering formulations of the $p-$center facility location problem and the discrete ordered median problem (see [5] and references therein). In this formulation, we first need to sort the distances between all objects in a non-decreasing order. Suppose that distances are denoted and ordered as $0 \leq d^1 < d^2 < \cdots < d^R = D$, where we denote the number of distinct pairwise distances by R. Furthermore, we define the additional binary decision variables w^r such that, for each $r = 1, \ldots, R$, w^r is 1 if and only if the overall smallest pairwise distance is at most d^r. Using the additional variables w^r, the covering linear formulation of MaxDP is as follows ($MaxDP_{cov}$):

$$\text{Max} \quad d^R + \sum_{r=1}^{R-1} (d^r - d^{r+1}) w^r \tag{6}$$

$$\text{s.t.} \quad (2) - (4), \tag{7}$$

$$x_{ic} + x_{jc} \leq 1 + w^r, \qquad \forall i,j \in V, i < j, c \in C, 1 \leq r \leq R : d_{ij} = d^r, \tag{8}$$

$$w^{r-1} \leq w^r, \qquad 2 \leq r \leq R, \tag{9}$$

$$w^r, x_{ic} \in \{0,1\}, \qquad \forall i \in V, c \in C, 1 \leq r \leq R. \tag{10}$$

The objective function (6) and the additional constraints (8) and (9) guarantee that the solution of the $MaxDP_{cov}$ provides a solution for the MaxDP. In particular, the w variable builds a vector: $w = (0, \overset{s}{\ldots}, 0, 1, \overset{R-s}{\ldots}, 1)$, with $0 \leq s \leq R$. This formulation is interesting, in one hand, due to its connections with some classical location problems and, on the other hand, $MaxDP_{cov}$ can be solved by any standard Integer Programming (IP) solver such as Gurobi, IBM Cplex, etc.

3 Heuristic Solution Method

Since the MaxDP is known to be NP-hard [4], we present a heuristic for solving the problem. One of the most successful heuristic methods is the Variable Neighborhood Search (VNS). Its concept was first introduced by Mladenović et al. [9]. Further, the VNS has been used for solving a large variety of (combinatorial) optimization problems for which this approach often provide high quality solutions (see, e.g., [2, 7], and references therein).

The VNS starts with an initial (feasible) solution as the incumbent solution. At each iteration, the algorithm generates diversification by making k random moves in a specific neighborhood of the incumbent solution. This move is known as the *shaking* step. The counter k can be started e.g., from 1 and is used to determine the diversifying range of the shaking step. Afterwards, a local search mechanism is used to improve the obtained solution. In this step, a set of designated neighborhoods is used. After termination of the local search, the objective value of the obtained solution is compared to the objective value of the incumbent solution. If the obtained objective value is not better, the process starts again with an intensified shaking step. Therefore the count k is raised by 1. The increment is done up to a predefined limit k_{max}. If the obtained objective value is better, the obtained solution becomes the incumbent solution and the process is restarted and the value of k is set down to 1. This process is repeated until a predefined termination criterion is fulfilled. We adopt this general framework for solving the MaxDP.

In particular, our VNS starts from a randomly generated solution that, in fact, consists in assigning (randomly) objects to the groups. Since the solution may not be feasible, due to violation of the balancing constraints, the objects are switched between different groups until achieving the feasibility. Then, the VNS uses Variable Neighborhood Descent (VND) as local search method. In a VND, the neighborhoods are used and explored in a consecutive order, i.e., if the local search finds no improvement in one neighborhood, then it switches to the next one. As soon as an improvement is found in any neighborhood, the procedure proceeds with the first one. In our design of VNS for the MaxDP, the VND may be applied on three different basic neighborhoods: *Insertion*, *Swap*, and *3-Chain*. Under the condition of preserving feasibility, these neighborhood structures are defined as follows:

- The neighborhood denoted by *Insertion* contains solutions obtained by moving one single object from its current group to another group.
- A solution in the *Swap* neighborhood is generated by swapping a single pair of objects belonging to different groups.
- A move in the *3-Chain* neighborhood is determined by three objects belonging to three different groups: Consider 3 objects u, v, w and their associated groups g_u, g_v, and g_w. The new solution is generated by moving objects in a circular way from one group to another one [2].

At each neighborhood, the result of the operation is accepted as a new (feasible) solution, if it leads to an improvement of the objective function value. Finally, we use the same procedure as [2], in order to admit the possibility of accepting solutions that do not lead to an improvement of the objective value (see [2] for more details).

4 Numerical Experiments

In order to evaluate the efficiency of the proposed VNS approach for solving the MaxDP, we carried out some numerical experiments on randomly generated instances. In this section, we present the test setting, the results of the experiments, and our observations. The results of the VNS is compared with the solutions provided by Gurobi 6.04 in the 64 bit version. The standard IP solver Gurobi has been used for solving the model $MaxDP_{cov}$. The algorithms are coded in Python 2.7 and ran on an Intel Celeron CPU G1620 using 4 GB RAM. In order to have a fair comparison, all experiments were done under same conditions.

We generated 3 test instances for each size 100, 200, 300, 400, and 500, i.e., in total 15 instances. On each instance, we set up 2 experiments for partitioning objects in either 4 or 10 groups. Furthermore, we set a time limit of $t_{max} = 1200$ s on the running time of each method (VNS as well as Gurobi). The parameter k_{max} (in the shaking operation of VNS) is set to 3 for all instances.

According to our observations, the 3-Chain neighborhood structure is computationally expensive in comparison to the Insertion and to the Swap neighborhoods. Hence, we excluded the 3-Chain neighborhood from our final set of experiments and we considered, for the VND, the following 2 different neighborhood structures:

- *VND-1*: Using only the Swap neighborhood.
- *VND-2*: Using the Insertion neighborhood followed by the Swap neighborhood.

The results are presented in Table 1. In this table, for each test instance of size (n), the optimum provided by Gurobi (*Obj.*) as well as the average optimal value (*aveObj.*), and the average gap (*Gap*) for each VNS method is presented.

According to the results, the VNS gives high quality solutions for small sized instances; in particular, when $n = 100$ and $g = 4$. The quality of the solution is

Table 1 The results for $g = 4$ and $g = 10$

n	$g = 4$					$g = 10$				
	Gurobi	VNS (VND-1)		VNS (VND-2)		Gurobi	VNS (VND-1)		VNS (VND-2)	
	Obj.	aveObj.	Gap	aveObj.	Gap	Obj.	aveObj.	Gap	aveObj.	Gap
100	0.7956	0.7956	0.0000	0.7956	0.0000	2.0699	1.9563	0.1136	1.5309	0.5390
200	0.5532	0.4962	0.0570	0.4886	0.0646	1.3134	0.7306	0.5828	0.4201	0.8933
300	0.3724	0.2591	0.1133	0.2090	0.1634	1.0932	0.3620	0.7312	0.1412	0.9520
400	0.3303	0.1315	0.1988	0.1052	0.2251	0.8757	0.2253	0.6504	0.0656	0.8101
500	0.3072	0.0886	0.2186	0.0614	0.2458	–	0.1190	–	0.0694	–

moderate for larger instances. However, an interesting point concerns the case of $n = 500$ and $g = 10$ for which, VNS manages to provide a feasible solutions (with a similar quality that it has for the other instances), but Gurobi fails to provide any solution.

The comparison of the two variants of VNS shows that the VNS using VND-1 outperforms the VNS using VND-2. In 23 out of 30 cases, using VND-1 leads to better results.

5 Conclusion

In this paper, we proposed a VNS approach for solving the Maximum Dispersion Problem. Based on the obtained numerical results, we observe that the VNS method leads to good results in small instances. In particular, VNS is able to provide integer solutions for all of the investigated test instances; however, for one of the instances with $n = 500$, the standard solver Gurobi fails to generate any solution within the predefined time limit. Future research avenue is quite wide. For example, we may investigate a more elaborate VNS e.g., by providing a more intelligent initialization procedure or by studying various neighborhood structures in order to explore the solution space in a more efficient way. The research in these directions is in progress and the results will be reported in future.

References

1. Baker, K.R., Powell, S.G.: Methods for assigning students to groups: a study of alternative objective functions. J. Oper. Res. Soc. **53**(4), 397–404 (2002)
2. Brimberg, J., Mladenović, N., Urošević, D.: Solving the maximally diverse grouping problem by skewed general variable neighborhood search. Inf. Sci. **295**, 650–675 (2015)
3. Erkut, E.: The discrete p-dispersion problem. Eur. J. Oper. Res. **46**(1), 48–60 (1990)
4. Fernández, E., Kalcsics, J., Nickel, S., Ríos-Mercado, R.Z.: A novel maximum dispersion territory design model arising in the implementation of the WEEE-directive. J. Oper. Res. Soc. **61**(3), 503–514 (2010)
5. Fernández, E., Kalcsics, J., Nickel, S.: The maximum dispersion problem. Omega **41**(4), 721–730 (2013)
6. Glover, F., Ching-Chung, K., Dhir, K.: A discrete optimization model for preserving biological diversity. Appl. Math. Modell. **19**(11), 696–701 (2010)
7. Hansen, P., Mladenović, N.: Variable neighborhood search. In: Glover, F., Knochenberger, G. (eds.) Handbook of Metaheuristics, pp. 145–184. Kluwer Academic Publisher (2003)
8. Martí, R., Gallego, M., Duarte, A.: A branch and bound algorithm for the maximum diversity problem. Eur. J. Oper. Res. **200**(1), 36–44 (2010)
9. Mladenović, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997)
10. Palubeckis, G., Karčiauskas, E., Riškus, A.: Comperative performance of three metaheuristic approaches for the maximally diverse grouping problem. Inf. Technol. Control **40**(4), 277–285 (2011)
11. Prokopyev, O., Kong, N., Martinez-Torres, D.: The equitable dispersion problem. Eur. J. Oper. Res. **197**(1), 59–67 (2009)

Part XIII
Optimization Under Uncertainty

Optimization of Modular Production Networks Considering Demand Uncertainties

Tristan Becker, Pascal Lutter, Stefan Lier and Brigitte Werners

Abstract In the process industry markets are facing new challenges: while product life cycles are becoming shorter, the differentiation of products grows. This leads to varying and uncertain product demands in time and location. As a reaction, the research focus shifts to modular production, which allow for a more flexible production network. Using small-scale plants, production locations can be located in direct proximity to resources or customers. In response to short-term demand changes, capacity modifications can be made by shifting modular units between locations or numbering up. In order to benefit from the flexibility of modular production, the structure of the network requests dynamic adaptions in every period. Subsequently, once the customer demand realizes, an optimal match between disposed production capacities and customer orders has to be determined. This decision situation imposes new challenges on planning tools, since frequent adjustments of the network configuration have to be computed based on uncertain demand. We develop stochastic and robust mixed-integer programming formulations to hedge against demand uncertainty. In a computational study the novel formulations are evaluated based on adjusted real-world data sets in terms of runtime and solution quality.

1 Introduction

Innovative, transformable production concepts implemented in standardized transportation iso-containers presently are in research focus in the process industry. Current market changes demanding short construction times motivate the usage of transformable plant designs. The changing dynamics imposed on the markets are

T. Becker (✉) · P. Lutter · B. Werners
Faculty of Management and Economics, Chair of Operations Research and Accounting, Ruhr University Bochum, Bochum, Germany
e-mail: tristan.becker@rub.de

S. Lier
Department of Mechanical Engineering, Laboratory for Fluid Separations, Ruhr University Bochum, Bochum, Germany
e-mail: lier@fluidvt.rub.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_55

characterized by shortened product life cycles, strong product differentiation and volatile product demands [2, 6]. Conventional plant designs in large scale, providing long time to markets and high investment risks are insufficient as reaction on volatile, uncertain markets. Hence, initial demonstration plants in transformable design were already developed, constructed and operated within several research projects like the EU funded F-Factory project or the CoPIRIDE project [4].

There exists a vast amount of literature on facility location problems and supply chain design. Besides general frameworks [5], there are also specific applications to the chemical industry [1]. Still, there is little literature on modular production network design, which represents an extended form of a facility location problem and is specifically characterized by the possibility of modular capacity shifts between locations. This work improves existing formulations [7] and provides an extension by stochastic and robust approaches in order to cope with uncertainty in modular production network configuration. A case study is conducted to demonstrate the value of information regarding uncertainty in comparison to a deterministic approach. The remainder of the paper is structured as follows. In Sect. 2 planning for modular production networks is further illustrated. Models for optimization of modular production networks and the approaches to uncertainty are outlined in Sect. 3. Finally, solution and economic performance of uncertainty approaches in modular production network design are evaluated in a computational study in Sect. 4 using real-world data.

2 Production Network Design Using Modular Plants

Modular production units in small scale enable new degrees of freedom in production network configuration. Using the mobility and scalability by numbering up or down containers, new opportunities regarding supply chain and network structure are offered. Production locations can be placed directly in customer‘s or resources‘ proximity, possibly reducing both the distance to the customer and resource. Modular plants can be shifted between production locations or easily adjusted in capacity over time in case of demand shifts. Time to markets are drastically reduced compared to conventional large scale plants, as standardized formats allow for rapid process development.

The scope of modular network design therefore is on the medium-term, as opposed to conventional facility location and supply chain design, which plan for a very long time horizon because of the apparent link to expensive investment decisions. Furthermore, modular network design represents a problem that is concerned with more detail, as the flexibility of modular plants allows for reaction to changing demand data in the short-term. A revision of the network design, such as closing a facility because of new demand data, is mostly impossible with large-scale plants, which is why supply chain design typically represents an erratic problem. In contrast, because of the associated flexibility, modular plants are well suited for decentralized production and changes during the planning period. Further advantages associated with

decentralized modular production include quick demand response times, lean production and low material stocks. This allows for a reduction of logistics and overall cost of producing specialty chemicals in comparison to centralized production. In order to benefit from transformable plants' flexibility, it is crucial to identify the most cost efficient production network. The next section describes the application of stochastic and robust optimization methods to modular production network design.

3 Modular Production Network Optimization Under Uncertainty

Using modular plants, as opposed to conventional large-scale plants, various location and relocation decisions as well as production allocation decisions have to be planned frequently, whenever new demand data becomes available. In each time period, the following decisions have to be made:

1. Acquire modular plants/divest modular plants
2. Open/close production locations
3. Move modular plants
4. Customer/capacity assignment

The modular network configuration problem [7] can be modeled as a mixed-integer linear program, which resembles an extended form of the facility location problem. Modular capacity shifts between locations, which represent the central characteristic of modular network design, are represented by constraints (1) and (2).

Decision variable γ_{jt} denotes the number of modules available at location j in period t. The number of modules moved from location j to location j' in period t is given by $m_{jj't}$. Finally, the amount of modules ordered in period t is represented by a_t. The first constraint (1) assures each location can only use the amount of modules that have been moved there in the current or previous periods. Constraint (2) ensures that modules which have been ordered are added to the modular hub which is denoted by location $j = 0$. They can then be moved out for production to open locations. Figure 1 depicts capacity shifts between locations as a result of a changing demand pattern over time.

$$\gamma_{jt} = \gamma_{jt-1} + \sum_{j' \in J} (m_{j'jt} - m_{jj't}) \quad \forall j \in J, t \in T \tag{1}$$

$$\gamma_{0t} = \gamma_{0t-1} + \sum_{j' \in J} (m_{j'0t} - m_{0j't}) + a_{t-1} \quad \forall t \in T \tag{2}$$

Clearly, there is a differing scope of time associated with each decision. The acquisition of new plants has to be decided in advance, since there is a lead time for new modular plants and production locations have to be prepared for the operation of modular plants. Further the removal, transportation and setup of modular plants shifting locations takes some time. In contrast, the customer/capacity assignment can

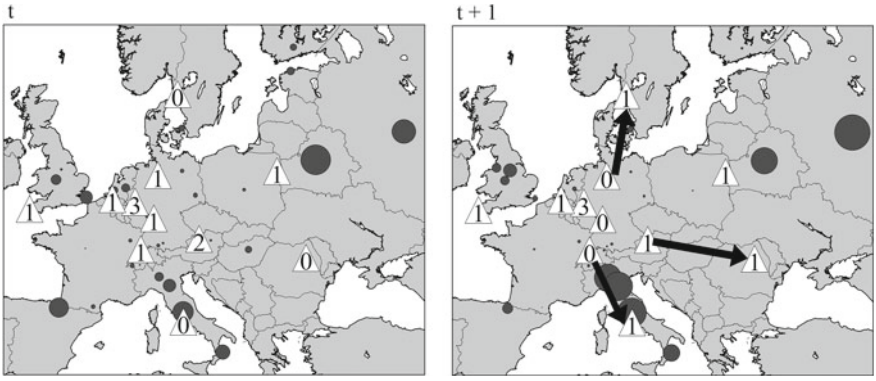


Fig. 1 Capacity shifts for modifying the network between two periods as a reaction to changed demands (*Triangle* production location, *circle* demand point)

be revised in the short term. In order to utilize the benefits of modular production, a cost-minimal network and production plan has to efficiently combine the interdependencies between decisions. The objective minimizes two types of cost: network and production cost. Network cost consist of fixed and variable location cost, module acquisition, movement and variable module cost. Production cost include raw material cost, transportation cost, production and shortage cost. The decisions can be partitioned into two stages:

- Stage 1: Minimize network configuration cost and uncertain production cost based on demand forecast
- Stage 2: Using the fixed network, satisfy demand with minimal production and shortage cost

Whereas the network has to be determined in stage one, a recourse with regard to customer/capacity matching is incorporated in stage two, once the demands have realized. The demand is assumed to be uncertain within a predetermined interval, while the interval is tighter bounded the closer the demand estimation is to realization. Given C customers and J potential locations, the goal is to find a network plan that minimizes total cost over the next periods based on uncertain demand data. To anticipate for uncertainty, the deterministic mixed-integer program is extended using two uncertainty approaches. The expected cost approach minimizes the expected value of the uncertain cost by using a set of scenarios resembling the demand distribution as closely as possible. Further a min-max regret approach is applied, using the same scenario set as possible realizations. The scenario set is obtained by sampling a large number of scenarios from the demand distribution and subsequently applying scenario reduction techniques [3].

4 Case Study

The economic and technical solution performance of the proposed deterministic and uncertainty models were evaluated on the basis of adjusted real-world data provided by an industrial partner. The computations were performed on instances of type *c4.xlarge* launched in the Amazon Elastic Compute Cloud. All of the models were implemented under utilization of the Gurobi Python Interface and subsequently solved with the 64-bit version of Gurobi 6.5 with default settings and 3600 s time limit. The real-world data set was used to generate a set of 36 test instances. To obtain test instances, the number of locations and customers as well as the demand level were systematically varied. The main difference between the approaches lies in the utilization of different information with regard to uncertainty. While the two uncertainty approaches explicitly incorporate a set of scenarios, the deterministic approach only considers the mean over all scenarios for every uncertain parameter. The network configuration solution of each of the approaches was used to evaluate the economic solution quality in a simulation study, where the total cost consisting of network and uncertain cost associated with the recourse in 10000 random demand realizations was computed. Figure 2 shows the distribution functions of cost for the case of 38 customers, 35 locations and a high demand level. The cost performance is clearly improved by using either uncertainty approach, whereas the preference for the expected cost or min-max regret solution depends on the risk attitude of the decision maker. Table 1 depicts the solution and cost performance of the different approaches.

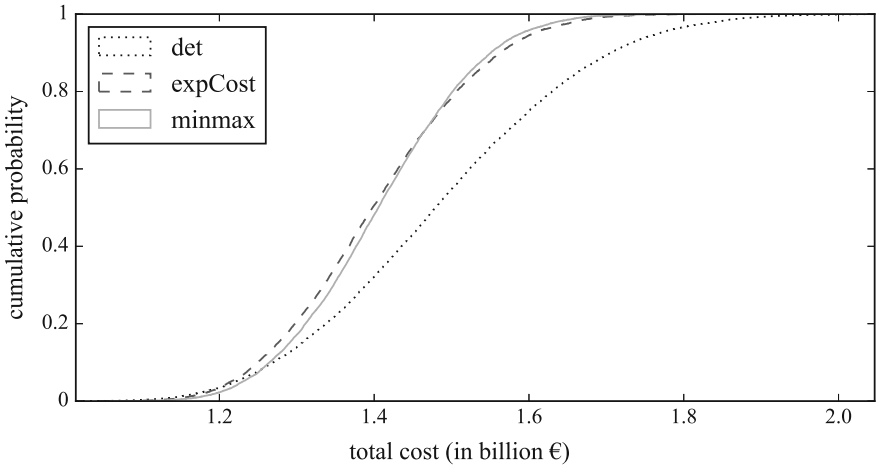


Fig. 2 Cumulative distribution function for each uncertainty approach

Table 1 Average solution and cost performance (cost standardized to 1 resp. column min.)

Model formulation	Sol. time (s)	Gap[a] (%)	Mean	Min	Max	Std.	VaR95
Deterministic	291.47	0.01	1.0505	1	1.1393	1.4756	1.1045
Expected cost	385.98	0.01	1	1.0256	1.0208	1.0899	1.0114
Min-max regret	1981.86	0.94	1.0031	1.0403	1	1	1

[a]Calculated as 100 * (Best bound − Best objective)/Best objective

5 Conclusion

Prior work on facility location problems provides a rich methodology for modeling complex issues and finding solutions to the resulting models. However, modular production network design has received little attention. In this study, the performance of three mixed-integer formulations for modular production network design under consideration of demand uncertainties has been investigated. An extensive computational study using an adjusted real-world data set has demonstrated the cost benefits of uncertainty approaches for modular production network design. However, the solution times, especially in case of the min-max regret approach, are not yet well suited for quick revisions in network planning. Future research should therefore focus on solution methodology for the modular production network planning problem.

References

1. Al-Othman, W.B.E., Lababidi, H.M.S., Alatiqi, I.M., Al-Shayji, K.: Supply chain optimization of petroleum organization under uncertainty in market demands and prices. Eur. J. Oper. Res. **189**(3), 822–840 (2008). doi:10.1016/j.ejor.2006.06.081
2. Buchholz, S.: Future manufacturing approaches in the chemical and pharmaceutical industry. Chem. Eng. Process. Process Intensif. **49**(10), 993–995 (2010). doi:10.1016/j.cep.2010.08.010
3. Dupacova, J., Gröwe-Kuska, N., Römisch, W.: Scenario reduction in stochastic programming. Math. Program. **95**(3), 493–511 (2003). doi:10.1007/s10107-002-0331-0
4. European Commission: Final Report for Flexible, Fast and Future Production Processes. Technical Report (2014)
5. Melo, M.T., Nickel, S., da Gama, F.S.: Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. Comput. Oper. Res. **33**(1), 181–208 (2006). doi:10.1016/J.Cor.2004.07.005
6. Shah, N.: Process industry supply chains: advances and challenges. Comput. Chem. Eng. **29**, 1225–1235 (2005). doi:10.1016/j.compchemeng.2005.02.023
7. Wörsdörfer, D., Lutter, P., Lier, S., Werners, B.: Optimized modular production networks in the process industry. In: Dörner, K., Ljubic, I., Pflug, G., Tragler, G. (eds.) Operations Research Proceedings 2015. Springer, Switzerland (2017). doi:10.1007/978-3-319-42902-1

Part XIV
Pricing and Revenue Management

Revenue Management Meets Carsharing: Optimizing the Daily Business

Justine Broihan, Max Möller, Kathrin Kühne, Marc Sonneberg and Michael H. Breitner

Abstract Carsharing is a transportation alternative that enables flexible use of a vehicle instead of owning it by paying trip-dependent fees. In recent years, this service denotes a considerable increase of new providers, which face an exponentially growing number of customers worldwide. As a consequence, rising vehicle utilization leads providers to contemplate revenue management elements. When focusing on station-based carsharing concepts, these are typically based on advance reservations. This makes them perfectly suitable for the application of demand-side management approaches. Demand-side management allows providers to optimize their revenues by accepting or rejecting certain trips. We respectively develop an optimization model for revenue management support. Based on an existing model of the hotel business, special consideration is drawn to carsharing related features. For instance, the implementation of a heterogeneously powered fleet allows providers to choose a certain limit of emissions to fulfill local requirements. We implement the mathematical model into the modeling environment GAMS using the solver Couenne. Conducted benchmarks show sensitivities under the variation of different input values, for example risk tolerances. In contrast to the often used first-come first-serve-principle, the results indicate the usefulness of the developed model in optimizing revenues of todays carsharing providers.

J. Broihan (✉) · M. Möller · K. Kühne · M. Sonneberg · M.H. Breitner
Leibniz Universität Hannover, Königsworther Platz 1, 30167 Hannover, Germany
e-mail: j.broihan@gmx.de

M. Möller
e-mail: max-moeller91@web.de

K. Kühne
e-mail: kuehne@iwi.uni-hannover.de

M. Sonneberg
e-mail: sonneberg@iwi.uni-hannover.de

M.H. Breitner
e-mail: breitner@iwi.uni-hannover.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_56

1 Introduction

Mobility is one major need of today's society. According to a survey by BMVI [2], 90% of the interviewed persons left a house at the reference day for various reasons whereas most of the distances (58%) were traveled by car. In large cities with highly developed public transportation systems, vehicle ownership is not always necessary and profitable. Furthermore, environmental awareness is a steadily increasing need [7] and raised the demand for carsharing in recent years [8]. However, carsharing providers are usually focused on profitability. Years of research and empirical knowledge point out that revenue management practices are an essential tool to successfully manage a company [1]. Accordingly, this paper addresses the following research questions:
RQ 1: How can revenue management practices be adapted to carsharing concepts?
RQ 2: How do the decision variables change, if local emission prerequisites vary?

2 Research Background and Optimization Model

Our literature review reveals that there is no published research on revenue management in combination with carsharing and a limited number of publications in combination with car rental. Respective models cover capacity management, pricing and reservation (Geraghty and Johnson [3]), assignment of vehicles to random customer requests by accepting or rejecting trips (Guerriero and Olivito [4]) and fleet distribution between rental stations, including capacity management at stations and the aspect of demand uncertainty (Haensel et al. [5]). Yet none of these models fully matches our focus on the combination of operator's risk aversion, customer satisfaction and demand uncertainty, which are deemed equally important aspects for the emergent business segment of carsharing. A more suitable model is introduced by Lai and Ng [6], who address demand uncertainty, operator's risk aversion, and customer satisfaction in the hotel business. Similarities between hotel and carsharing sectors include the availability of rooms or vehicles, the parallels in booking processes and the possibility of reservation purchase. We therefore transfer and adapt their model to suit our carsharing application.

To do so, several assumptions are necessary. The developed model considers different time frames with an interval duration of 3 h. Thus, a total of eight time frames per day result. Every started time frame must be paid entirely by the customer. A trip duration limit of 24 h is set. The revenue can be set individually per time frame by the provider. When the demand is low, the resulting revenue should be low as well, whereas the revenue increases with rising demand. A customer is able to make a reservation for a vehicle in advance. At the beginning and the end of the observation period all vehicles must be available. To allow for overnight trips, such bookings are divided into two bookings.

Table 1 Parameters—initial solution

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
$C_{1,1}$	25	$C_{1,2}$	13	$C_{2,1}$	20	$C_{2,2}$	10
CO_{max}	2,184 g	E_1	3,302 g	E_2	0 g	λ	1
p_1	1/3	p_2	1/3	p_3	1/3	$w_{i,j}$	1

The optimization model considers six indices. The indices $i = \{1, \ldots, T-1\}$ and $j = \{2, \ldots, T\}$ indicate the starting and ending time frame of the renting period. $k = \{2, \ldots, T-1\}$ represent any time in the observation period while $s = \{1, \ldots, S\}$ is the amount of several demand scenarios. We specify three scenarios with a low, middle and high demand level. The stations to be optimized are given by $z = \{1, \ldots, Z\}$ and the different vehicle types in terms of propulsion methods are given by $t = \{1, \ldots, N\}$. Vehicle type 1 represents a diesel-engined vehicle system, whereas vehicle type 2 is electrically powered. To limit the number of vehicles at any station, $C_{z,t}$ is the capacity at station z for each vehicle type t. A threshold concerning ecological needs is represented by CO_{max}, the maximum average admissible amount of CO_2-emissions over the whole fleet. The emission of the individual vehicle types, based on the propulsion method, is given by the parameter E_t. λ is a trade-off factor between expected revenue and deviation that gives the risk aversion of the management. The probability of a scenario is represented by p_s. A booking with starting and ending time i and j in scenario s delivers a revenue $R^s_{i,j}$. The corresponding demand at station s is given by $U^s_{i,j,z}$. $w_{i,j}$ is a parameter that weights the number of bookings with starting and ending time i and j. If $w_{i,j}$ is low, more bookings with the corresponding starting and ending times are satisfied. Finally, the decision variable $x_{i,j,z,t}$ provides the total number of accepted bookings for vehicle type t at station z with starting and ending time i and j. Due to the optimization of the operational planning level of a carsharing organization, costs for stations and vehicles are not considered. The values for the external parameters, which are obtained in corporation with a carsharing organization are given in Table 1.

$$
\begin{aligned}
\text{Max} \quad & \sum_{s=1}^{S} \left(p_s \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \sum_{z=1}^{Z} \sum_{t=1}^{N} (j-i) R^s_{i,j} x_{i,j,z,t} \right) \\
& - \lambda \sum_{s=1}^{S} \left(p_s \left| \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \sum_{z=1}^{Z} \sum_{t=1}^{N} (j-i) R^s_{i,j} x_{i,j,z,t} \right. \right. \\
& \left. \left. - \sum_{s=1}^{S} \left(p_s \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \sum_{z=1}^{Z} \sum_{t=1}^{N} (j-i) R^s_{i,j} x_{i,j,z,t} \right) \right| \right) \\
& - \sum_{s=1}^{S} \left(p_s \sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \sum_{z=1}^{Z} \sum_{t=1}^{N} w_{i,j} | U^s_{i,j,z} - x_{i,j,z,t} | \right)
\end{aligned}
\tag{1}
$$

$$\text{s.t.} \quad \sum_{i=1}^{k-1} \sum_{j=k+1}^{T} x_{i,j,z,t} + \sum_{j=k+1}^{T} x_{k,j,z,t} \leq C_{z,t} \quad \forall k, z, t \tag{2}$$

$$\sum_{j=2}^{T} x_{1,j,z,t} \leq C_{z,t} \quad \forall z, t \tag{3}$$

$$\sum_{t=1}^{N} x_{i,j,z,t} \leq \max\{U_{i,j,z}^{s}\} \quad \forall i, j, z, \tag{4}$$

$$\frac{\sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \sum_{z=1}^{Z} \sum_{t=1}^{N} (j-i) E_t x_{i,j,z,t}}{\sum_{i=1}^{T-1} \sum_{j=i+1}^{T} \sum_{z=1}^{Z} \sum_{t=1}^{N} (j-i) x_{i,j,z,t}} \leq CO_{max} \tag{5}$$

$$\sum_{i=1}^{T-1} \sum_{z=1}^{Z} x_{i,8,z,2} = 0 \tag{6}$$

$$x_{i,j,z,t} \geq 0 \quad \forall i, j, z, t \tag{7}$$

$$1 \leq i < j \leq T \quad \forall s \in \Omega \tag{8}$$

The objective function (1) consists of four terms to maximize the daily revenues of the carsharing provider. The first term of this function maximizes the expected revenue in dependence of the occurrence of a certain scenario s. The average absolute deviation of the revenue is subtracted in the second term and is calculated by the absolute value of the difference of actual and expected revenue. The absolute deviation of the demand is subtracted in the third term and is calculated by the absolute value of the difference of demand and number of accepted bookings. Constraint (2) is the capacity restriction for the vehicle types and secures that the number of rented vehicles does not exceed the fleet size. Constraint (3) ensures that the number of accepted bookings does not transcend vehicles available at the beginning of the observation period. According to constraint (4), the number of accepted bookings must be smaller than the maximum demand of all scenarios. A maximum level of the CO_2-emissions is expressed in constraint (5) and secures that an average emission of all vehicles within the fleet is not higher than certain thresholds. To recharge electric vehicles, Eq. (6) guarantees that the number of accepted bookings of vehicle type 2 in time frame eight is equal to zero to ensure the recharging process of the electric vehicles, to be available at the beginning of an operating day. Furthermore, the number of accepted bookings must not be negative (7) and (8) specifies the validity range of starting and ending times frames.

3 Results, Sensitivities and Benchmarks

In this section, we present the results which are obtained by solving the mathematical model from Sect. 2 using GAMS 24.7.1 and the solver COUENNE 0.5 with a preset gap of 0%. Table 2 shows the number of accepted bookings for every combination of starting time and end of rental for station 1 and 2 with respect to propulsion method. In time frame 3, 18 diesel-engined vehicles are rented at station 1. 13 of these rentals are returned at time frame 5 and the remaining five vehicles end at time frame 8. The objective function value amounts to 6,831.74 €. Compared to the often used

Table 2 Accepted bookings—initial solution

Starting time frame	Station 1														Station 2													
	Diesel							Electric							Diesel							Electric						
	Ending time frame																											
	2	3	4	5	6	7	8	2	3	4	5	6	7	8	2	3	4	5	6	7	8	2	3	4	5	6	7	8
1	3	3	1	–	–	–	–	11	2	–	–	–	–	–	6	4	2	–	–	–	–	10	–	–	–	–	–	–
2		3	–	–	–	–	–			–	–	–	–	–		–	–	–	–	–	–		–	–	–	–	–	–
3			–	13	–	–	5			–	–	–	11	–			–	7	–	–	7			–	–	–	10	–
4				–	–	–	6				–	–	2	–				–	–	–	4				–	–	–	–
5					–	–	1					–	–	–					–	–	2					–	–	–
6						–	13						–	–						–	7						–	–
7							–							–							–							–

first-come first-serve-principle, the presented model increases the objective function value by more than 49% (2,271.64 €) per day. The explanation for this significant difference lies is the improved resource utilization: profitable trips take precedence over less profitable or short term reservations. The initial solution comprises overall 339 rented time frames. 224 are served by diesel-engined vehicles and 115 by electric vehicles. 184 rented time frames are operated at station 1 whereas the remaining 155 are operated at station 2.

In order to derive the model sensitivities we reduce the parameter for the maximum CO_2 emission across the whole fleet to 1,000 g per time frame. This equals a reduction by approx. 55%. As a consequence, a decrease in the number of rented time frames for diesel-engined vehicles is observed. The results are presented in Table 3. A decrease in rented time frames of vehicle type 1 by 175 to 49 with regard to both stations is obtained. The objective function value decreases to 3,032.29 € which is, compared to the initial solution, a reduction of approx. 44% (3,799.45 €). Compared to the first-come first-serve-principle a 69% (1,241.98 €) improvement in the CO_2 low case can be achieved using the presented revenue management model.

Table 3 Accepted bookings—sensitivities and benchmarks

CO_2 low	Station 1														Station 2													
	Diesel							Electric							Diesel							Electric						
Starting time frame	Ending time frame																											
	2	3	4	5	6	7	8	2	3	4	5	6	7	8	2	3	4	5	6	7	8	2	3	4	5	6	7	8
1	–	–	–	–	–	–	–	11	2	–	–	–	–	–	–	–	–	–	–	–	–	10	–	–	–	–	–	–
2		–	–	–	–	–	–		–	–	–	–	–	–		–	–	–	–	–	–		–	–	–	–	–	–
3			9	–	–	–	–			–	–	–	11	–			9	8	–	–	–			–	–	–	10	–
4				–	–	–	–				–	–	2	–				–	–	–	–				–	–	–	–
5					–	–	–					–	–	–					–	–	–					–	–	–
6						5	1						–	–						4	2						–	–
7							–							–							–							–

4 Discussion and Conclusions

The objective of this paper was to optimize the daily revenue of a carsharing organization. An existing mathematical model to optimize the room occupancy of hotels was adapted to station-based carsharing. This was possible through similarities between the operating modes of both business segments. The resulting model allows to implement differently structured networks with regards to stations and vehicles. To fulfill (future) local prerequisites in terms of emissions, a CO_2 threshold over the average fleet can be set. This results in an assignment of differently powered vehicles to the existing stations without exceeding the predefined threshold. To demonstrate the general functionality and the influence of the parameter modifications with regards to emissions, we used the two extrema of possible propulsion methods, diesel-engined and electrically powered vehicles. In addition, we assume 0 g/km CO_2-emission for the electric vehicles. Future research should address certain limitations of our approach. Possible enhancements include the creation of shorter time frames and a minute- and/or kilometer-based billing. Additionally, the charging process can be optimized by allowing charging as needed rather than at the end of a period. To conclude, our developed model shows the applicability of revenue management to optimize the daily business of station-based carsharing services operating with heterogeneous fleets.

References

1. Bitran, G., Caldentey, R.: An overview of pricing models for revenue management. Manuf. Serv. Oper. Manage. **5**(3), 203–229 (2004)
2. BMVI (Bundesministerium für Verkehr und digitale Infrastruktur): Mobilität in Deutschland (MiD). http://www.bmvi.de/SharedDocs/DE/Artikel/G/mobilitaet-in-deutschland.html (2015). Accessed 20 Oct 2015
3. Geraghty, M., Johnson, E.: Revenue management saves national car rental. Interfaces **27**(1), 107–127 (1997)
4. Guerriero, F., Olivito, F.: Revenue models and policies for the car rental industry. J. Math. Model. Algorithms Oper. Res. **13**(3), 247–282 (2014)
5. Haensel, A., Mederer, M., Schmidt, H.: Revenue management in the car rental industry: a stochastic programming approach. J. Revenue Pricing Manage. **11**(1), 99–108 (2012)
6. Lai, K., Ng, W.: A stochastic approach to hotel revenue optimization. Comput. Oper. Res. **32**, 1059–1072 (2005)
7. Schack, K., Gellrich, A.: Umweltbewusstsein in Deutschland 2014. Ergebnisse einer repräsentativen Bevölkerungsumfrage. Bundesministerium für Umwelt, Naturschutz, Bau und Reaktorsicherheit (2015)
8. Statista GmbH: Anzahl der stationsbasierten Carsharing-Fahrzeuge in Deutschland in den Jahren 2009 bis 2015. http://de.statista.com/statistik/daten/studie/219139/umfrage/anzahl-der-carsharing-fahrzeuge-in-deutschland/ (2015). Accessed 19 Nov 2015

Exogenous Capacity Changes in Airline Revenue Management: Quantifying the Value of Information

Daniel Kadatz, Natalia Kliewer and Catherine Cleophas

Abstract In airline revenue management, capacity is usually assumed to be fixed. However, capacity changes are common in practice. This contribution quantifies the value of information when systematically considering possible capacity changes in revenue optimization. It solves a stochastic model that anticipates capacity changes, given different levels of information. A computational study compares solution approaches with respect to the resulting revenue, seat load factor, and denied boarding.

1 Introduction

Classic airline revenue management controls bookings for capacitated, perishable products to maximize revenue from ticket sales. Most models assume that capacity is not only limited, but given in advance and fixed (compare [6, p. 3]). However, this does not always hold in practice. Analyzing data provided by Lufthansa German Airlines, we found that up to 66% of all flights were affected by at least one capacity change. Some contributions propose adjusting capacity to demand variation, but existing research rarely considers capacity changes beyond that motivation.

While we call changes induced to compensate demand variation *endogenous*, we call changes caused beyond revenue management *exogenous*. Reasons for exogenous changes include technical defects, crew planning, special sales, bad weather conditions, and strikes.

D. Kadatz (✉) · N. Kliewer
Information Systems, Freie Universität Berlin, Garystr. 21, 14195 Berlin, Germany
e-mail: daniel.kadatz@fu-berlin.de

N. Kliewer
e-mail: natalia.kliewer@fu-berlin.de

C. Cleophas
Advanced Analytics, RWTH Aachen University, Kackertstr. 7, 52072 Aachen, Germany
e-mail: catherine.cleophas@ada.rwth-aachen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_57

Within the scope of this contribution, we assume different levels of information on possible capacities, the timing, and the probability of changes to be given. As our primary contribution, we examine the value of this information. In the course of such analysis, we consider a formal stochastic optimization model as introduced in [3].

2 State of the Art

Most existing revenue management contributions endogenously adjust capacity in response to demand variation. Related concepts are termed demand driven dispatch [1], demand driven swapping [2], or dynamic capacity management [5].

One of the first to propose using aircraft families are [1]. They claim a resulting revenue improvement of 1–5%. Bish et al. [2] only allow swaps of two aircrafts within one aircraft family. Wang and Regan [8] also study aircraft swaps as an extension of leg-based revenue management, albeit from a perspective of continuous time. To adjust capacity while maximizing revenue, [4] proposes EMSR-d. The approach by [5] allows for continuously adjusting capacity given a dependent demand model. Vulcano and Weil [7] consider the joint optimization of virtual capacities and bid prices. Our study employs the same demand factors, customer generation process, and number of demand streams.

While those approaches propose capacity flexibility, we oppose to *capacity uncertainty*. To the best of our knowledge, so far only two contributions integrate exogenous capacity changes in revenue maximization. Wang and Regan [8] introduce the concept of capacity uncertainty to support their framework of repeated aircraft swaps under the assumption of continuous time. In contrast, [3] propose a formulation based on a discrete-time revenue management approach. Both decompose the problem of capacity uncertainty into a time period before a capacity change can occur and after that. However, [3] expand the framework by allowing a more realistic number of capacity change variables. While [8] limit possible capacity changes to a single swap [3] allow for any timing of change and an arbitrary number of potential changes. The model by [3] provides the theoretical background of our work. In contrast to [3] we allow for multiple changes over the booking horizon and re-optimize after every change.

3 Model and Solution Approaches

The model considers optimizing the number of tickets per class over a single leg and compartment. Customers do not cancel their reservation, so denied boardings only occur when capacity is overestimated.

The booking horizon is characterized through time slices $t \in T := \{\hat{t}, \dots, 0\}$, where $\hat{t}$ denotes the beginning of the booking horizon and 0 signifies the time of departure. Revenue r_f is fixed per fare class $f \in F$. For each time slice t and fare

class f, expected demand is indicated by $d_{ft} \in \mathbb{N}$. In the model, $k_a \in \mathbb{N}$ denotes the cost of the ath denied boarding, $a \in \{1, \ldots, A\}$. Whether a denied boarding occurs is indicated by $e_a \in \{0, 1\}$. Denied boarding cost increase, i.e., $k_1 \leq \cdots \leq k_A$.

Capacity changes are described by three characteristics: The resulting *capacity*, the *time* in the booking horizon when capacity is updated, and the *probability* of a particular change. The model uses scenarios to describe these characteristics. Each combination of capacity $c^s \in \mathbb{N}$ and time $t^s \in T$ describes a scenario $s \in S$ with a respective probability p^s, where $\sum_{s \in S} p^s = 1$.

A decision variable for a *global strategy* $x_{ft} \in \mathbb{N}$ defines the number of tickets to offer in fare class f at time t. This strategy is executed until scenario s announces a possible change to capacity c^s at time t^s. This triggers the *scenario-based strategy* $x^s_{ft} \in \mathbb{N}, f \in F, t \in T, t > t^s$, which defines the number of tickets offered in fare class f at time $t \leq t^s$.

The objective (1) is to maximize the number of sold tickets x_{ft} and x^s_{ft} multiplied by revenues r_f and to simultaneously minimize the denied boarding cost $\sum_{a \in |A|} \sum_{s \in S} e^s_a \cdot k_a$. The capacity restriction (2) ensures that the sum of sold tickets x_{ft} and x^s_{ft} exceeding an aircraft's capacity c^s results in denied boardings, denoted by variable e^s_a. The demand restrictions (3) ensure that sold tickets can not exceed demand d_{ft}.

$$\underset{x_{ft},\, x^s_{ft},\, e^s_a}{\text{maximize}} \sum_{s \in S} p^s \left(\sum_{f \in F} r_f \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{a=1}^{A} \sum_{s \in S} e^s_a \cdot k_a \right) \tag{1}$$

$$\text{s.t.} \sum_{f \in F} \left(\sum_{t=\hat{t}}^{t^s+1} x_{ft} + \sum_{t=t^s}^{0} x^s_{ft} \right) - \sum_{a=1}^{A} e^s_a \leq c^s \quad \forall\, s \in S \tag{2}$$

$$x_{ft} \leq d_{ft} \quad \text{and} \quad x^s_{ft} \leq d_{ft} \quad \forall\, s \in S,\ f \in F, t \in T \tag{3}$$

$$x_{ft},\, x^s_{ft} \in \mathbb{N} \quad \forall\, s \in S,\ f \in F, t \in T$$
$$e^s_a \in \{0, 1\} \quad \forall\, s \in S, a \in \{1, \ldots, A\}$$

Given information on capacity, timing, and scenario probabilities, this stochastic problem can be solved to optimality. We denote this approach as C-s/P-s/T-s. The stochastic model collapses to a deterministic version when assuming a single scenario with probability 1.0.

For different levels of information, alternative solution approaches are conceivable. Each is updated to account for new information when a capacity change is announced.

C*—Perfect foresight This upper bound solves the deterministic model given perfect foresight of the actual final capacity.

C-s/P-s/T-s This solution approach uses the full spectrum of scenario information. Thus C-s/P-s/T-s takes all capacities, probabilities and timings into account.

C-s/P-uni/T-s Without information of scenario probabilities, this approach creates as many scenarios as there exist combinations of potential capacities and change times. It assigns each scenario a uniform probability.

C-s/P-s/T-0 Without information on changes' timing, this approach creates as many scenarios as there exist possible capacities, and assigns each a given probability. It assumes that the final capacity is revealed at departure.

C(P-max) This approach assumes that the most probable scenario will occur. Therefore, it only solves the problem for the capacity resulting from that scenario.

C(P-s) This approach works similarly to C(P-max), but parameterizes the single scenario using the arithmetic mean of all possible capacities weighted by their probability. It does not require information on timing.

C-s/P-uni/T-0 Without information on change's timing or probabilities, this approach creates one scenario for each possible capacity. It assumes that each scenario is equally probable and that capacity is revealed at departure with equal probability.

C(P-uni) This approach does not require information on changes' timing or probability. It works similarly to C(P-s), but parameterizes the single scenario as the arithmetic mean of all capacities.

C-min Without requiring information on probabilities and timings, this approach solves the problem for the scenario given the minimum capacity.

C-ini Without any information on changes, this approach considers only the initially announced capacity. After every capacity update, it solves the model based on the new capacity.

4 Computational Study

The computational study considers three classes with revenues per ticket respective $r_1 = 200, r_2 = 150$ and $r_3 = 100$. Denied boarding cost are modeled as exponentially increasing by a factor. The first denied boarding costs 201, exceeding the highest fare. The second denied boarding cost is $k_2 = k_1 \times 1.1 = 201 \times 1.1 \approx 221$ and so on.

To generate patterns of request arrivals over the booking horizon, we analyzed a set of historical booking data provided by Lufthansa. We emphasize the role of request arrival timings as its interaction with the timing of capacity changes is of high interest for our research. The resulting triangular distributions are used as input for the customer generation via a non-homogenous Poisson process.

We assume demand for three fare classes to be independent and set the average number of customers by multiplying the flight's initial capacity by a demand factor in $\{0.9, 1.2, 1.6, 1.8\}$. Furthermore, we implement three distributions of demand over fare classes: {(50-25-25), (25-50-25), (25-25-50)}. E.g., (50-25-25) determines that 50% of all customers request the most expensive class, while 25% request each of the other classes.

Results are given for 1,000 stochastic demand streams for every combination of market, initial capacity, demand factor, and distribution of demand over fare classes.

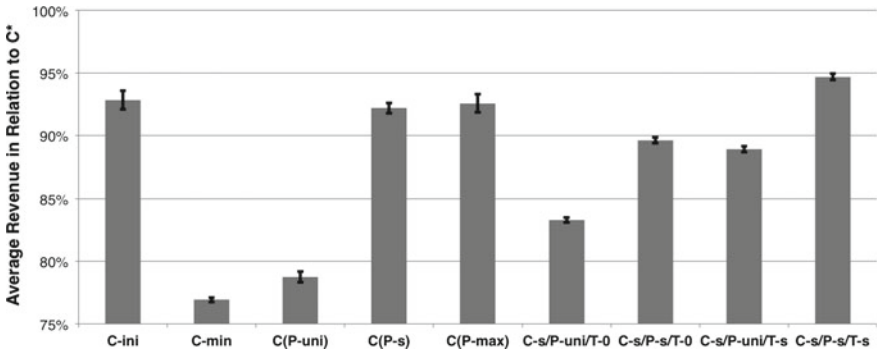


Fig. 1 Average revenue in relation to C* (*grey bars*) and 95% confidence interval (*black lines*)

We calibrate capacities based on Lufthansa data, focusing on the economy compartment. For every capacity change, the empirical data set records the timing and the resulting capacity.

We always put revenue in relation to the upper bound delivered by C*. Figure 1 shows revenues obtained with each solution approaches. C-s/P-s/T-s performs best with an average of almost 95% of C*. This is an average advantage of 1.9% over C-ini. Also, the confidence interval is small, which implies a more robust approach.

In contrast, C-min results in notably less revenue. Although C(P-max) performs almost as well as C-ini, we can neglect this solution approach, as it imitates C-ini in the majority of instances. While C(P-s) does not perform as well as C-ini, it seems to be a good heuristic for high change probabilities. When comparing C-s/P-s/T-0 and C-s/P-uni/T-s with C-s/P-s/T-s, we can state that knowing capacity changes' probabilities is more valuable than knowing their timing. Figure 2 explains the revenue differences by illustrating the average seat load factor and denied boardings. The seat load factor is the number of tickets sold divided by final capacity, expressed as per-

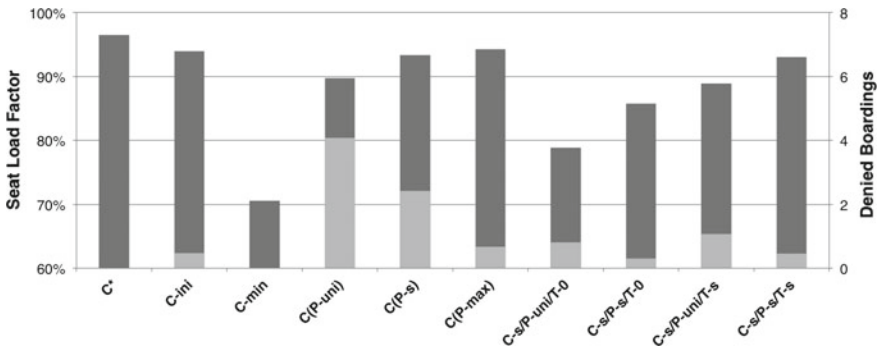


Fig. 2 Average seat load factor (*dark bar*) and average denied boardings (*light bar*)

centage. The maximum seat load factor is 1.0, as exceeding tickets result in denied boardings. Denied boardings are indicated in absolute numbers.

As expected, the *oracle* solution C* has the highest average seat load factor and never induces denied boardings. C-min also never causes denied boardings, but its seat load factor is lower. On average, C(P-uni) produces the most denied boardings, as it overestimates scenarios with a high capacity. Both C-s/P-s/T-s and C-ini lead to few denied boardings. However, C-ini's seat load factor is almost 1% higher than that of C-s/P-s/T-s. Nevertheless, C-s/P-s/T-s generates more revenue on average, as it sells more high value tickets.

5 Conclusion

This contribution considered the value of information when modelling stochastic capacity in a leg-based revenue optimization model. While some heuristics performed close to optimality, taking all information into account could generate an average 1.9% revenue advantage. When not all information can be considered, knowing a capacity change's probability is more beneficial than only knowing its timing. Future research requires further analyses with regard to aircraft sizes, demand level, demand mix, and forecasting errors.

Acknowledgements We thank Lufthansa German Airlines for access to empirical data.

References

1. Berge, M., Hopperstad, C.: Demand driven dispatch: a method for dynamic aircraft capacity assignment, models and algorithms. Oper. Res. **41**(1), 153–168 (1993)
2. Bish, E., Suwandechochai, R., Bish, D.: Strategies for managing the flexible capacity in the airline industry. Naval Res. Logistics (NRL) **51**(5), 654–685 (2004)
3. Büsing, C., Kadatz, D., Cleophas, C.: Considering exogenous capacity changes in airline revenue management: models, algorithms and computations. Working Paper (2015)
4. De Boer, S.: The impact of dynamic capacity management on airline seat inventory control. J. Revenue Pricing Manage. **2**(4), 315–330 (2004)
5. Frank, M., Friedemann, M., Mederer, M., Schroeder, A.: Airline revenue management: a simulation of dynamic capacity management. J. Revenue Pricing Manage. **5**(1), 62–71 (2006)
6. Talluri, K., van Ryzin, G.: The Theory and Practice of Revenue Management. Springer (2004)
7. Vulcano, G., Weil, A.: Joint optimization of virtual capacities and bid-prices for revenue management. Working Paper (2014)
8. Wang, X., Regan, A.: Dynamic yield management when aircraft assignments are subject to swap. Transp. Res. Part B Methodol. **40**(7), 563–576 (2006)

Integrated Planning of Order Capture and Delivery for Attended Deliveries in Metropolitan Areas

Charlotte Köhler, Magdalena A.K. Lang, Catherine Cleophas and Jan Fabian Ehmke

Abstract The ongoing boom in e-commerce increases the importance of profitable and customer-oriented delivery services. Particularly in metropolitan areas, the high population density offers great potential for e-commerce, while uncertain demand and traffic conditions increase planning uncertainty. This contribution focuses on e-commerce delivery fulfillment (e-fulfillment) for attended last-mile delivery services in metropolitan areas. As the customer needs to be present for deliveries of groceries, for example, a service time window has to be agreed upon already when a customer's order is accepted. We consider service time windows as a scarce resource and as the critical interface between order capture and order delivery. To optimally utilize this scarce resource, we propose combining concepts of revenue management and vehicle routing to extend tactical and operational planning for e-fulfillment. We define the research problem and provide a perspective on integrated planning for attended deliveries. Furthermore, we present the design of a virtual laboratory to support benchmarking in e-fulfillment research. To ensure realistic experimental settings, we plan to incorporate real-world data provided by a major e-grocery in Germany.

1 Introduction

With two-digit growth rates predicted for e-commerce revenues [2], profitable and customer-oriented services gain importance. In case of *attended deliveries*, e.g., of

C. Köhler (✉) · J.F. Ehmke
Business Analytics, European University Viadrina, Große Scharrnstraße 59,
15230 Frankfurt (Oder), Germany
e-mail: koehler@europa-uni.de

J.F. Ehmke
e-mail: ehmke@europa-uni.de

M.A.K. Lang · C. Cleophas
Advanced Analytics, RWTH Aachen University, Kackertstr. 7, 52072 Aachen, Germany
e-mail: magdalena.lang@ada.rwth-aachen.de

C. Cleophas
e-mail: catherine.cleophas@ada.rwth-aachen.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_58

perishable food, the customer has to be present at the time of delivery [4]. Therefore, companies and customers need to agree on service time windows already during order capture. This avoids costly delivery failures, but also reduces planning degrees of freedom. Especially in metropolitan areas, retailers have to rise to the challenge of successfully planning attended delivery services under uncertain demand and traffic conditions. At the same time, customers expect on-time delivery within tight time windows. Together with strong competition and potentially low profit margins, these challenges lead to a high failure rate of business concepts and motivate new delivery service approaches [1].

The process of e-commerce delivery fulfillment (e-fulfillment) can be structured in three steps: order capture and promise, order sourcing, and order delivery [3]. *Order capture* aims to maximize the number or value of accepted orders. Sophisticated planning methods for order capture require adequate demand forecast and market segmentation techniques. Revenue management uses such techniques to maximize revenues by selling units of the same resource to different customers at different prices. *Order sourcing* assembles the accepted orders, whereas *order delivery* minimizes delivery costs given accepted orders. Finding a cost-minimal route for a given fleet of vehicles while considering time window constraints can be modeled as the vehicle routing problem with time windows (VRPTW). To make attended last-mile deliveries in metropolitan areas more reliable, the VRPTW can be extended to consider time-dependent travel times [6, 7].

In contrast to their independent and sequential handling, we propose to integrate the planning of order capture and order delivery. This can be realized by combining methods of revenue management and vehicle routing. We consider service time windows as a scarce resource and as the critical interface between order capture and delivery. As such, it should be optimally utilized to maximize the profitability of e-fulfillment.

This contribution focuses on identifying the underlying research problems and discussing the idea of our approach. Additionally, we present the design of an experimental laboratory to benchmark methods for current and future concepts of e-fulfillment. We will calibrate the embedded simulation system with transactional data from a major e-grocery in Germany to ensure realistic experimental settings.

2 Related Literature

Several authors already approach integrating order capture and delivery aspects for e-fulfillment. They consider delivery costs, capacities, or revenues when accepting customer requests. In the following, we shortly review recent contributions and derive the research gap we want to target.

Some references emphasize the importance of time window allocation to increase cost-efficiency and reliability of attended home deliveries. Campbell and Savelsbergh [4] apply simple incentive schemes to attract customers to specific time slots. They strive for cost-efficient delivery through short travel times and try to increase

the number of accepted orders. Alternatively, Agatz et al. [1] control the availability of time slots per zip code to minimize expected delivery costs. Ehmke and Campbell [6] compare order acceptance mechanisms to maximize the number of accepted orders while ensuring reliability of delivery. They include time-dependent travel time information reflecting congestion in rush-hour time windows. These references consider both expected transportation costs and number of accepted orders, but they do not account for order value.

Revenue management considers the impact of an order's contribution to overall revenue. Quante et al. [9] review and categorize models and planning systems applicable for integrating revenue management into the general demand fulfillment process. Yang et al. [10] present a joint model for e-fulfillment of attended deliveries, concentrating on controlling delivery fees. Following a similar idea, Klein et al. [8] apply a differentiated time window pricing approach to maximize revenues. The direction of these contributions is similar to our approach. However, we argue that the span of delivery pricing provides little leverage compared to differences in order value within heterogeneous demand.

This paper builds on the work of Cleophas and Ehmke [5], who introduce an iterative order value-based e-fulfillment process. In the model, service time windows are fixed and their availability is flexibly controlled based on order values. A given fleet size determines the transport capacities. The planning process starts with a demand forecast for each combination of time window and delivery area. Additionally, orders are assigned to discrete value buckets. The forecast serves as input for an initial route planning, which determines transport capacities per time window and delivery area by preferring high-value orders. Order acceptance considers the value of actual orders as well as the given transport capacities. Once the full order set is determined, a final, cost-efficient routing prepares delivery.

The contribution calls for further research in regard to two main aspects. First, a distinct, sequential application of revenue management and vehicle routing is likely inferior to fully integrated planning. Furthermore, the computational study is based on a rather simple simulation approach with synthetic order and delivery data. To allow for a more realistic setting, we outline an experimental laboratory including complex simulations and empirical data to benchmark different process variants.

3 An Integrated Planning Approach to E-Fulfillment

Our research focuses on the conditions and effects of different degrees of integration between order capture and order delivery. Considering delivery time windows as a scarce resource, we control time slot allocation on order acceptance through revenue management and vehicle routing techniques. We aim to maximize overall revenue from order values while supporting cost-efficient order delivery.

In a first step, we consider the information necessary to integrate the two tasks and the planning methods that need to be extended. For example, order capture can include preliminary capacity information from vehicle routing to determine fulfill-

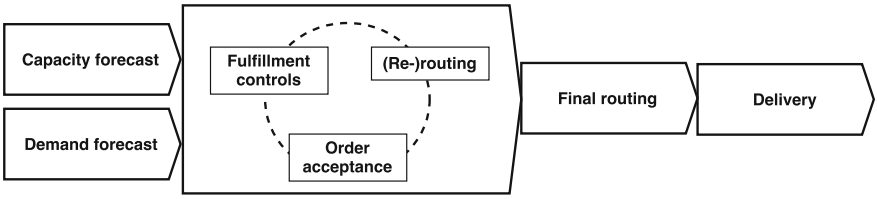


Fig. 1 Example of an integrated e-fulfillment process

ment controls. Vehicle routing can be applied on forecasted orders, taking their value into account.

Figure 1 exemplifies an adapted, more integrated process derived from the iterative approach of Cleophas and Ehmke [5]. Based on historical sales, a demand forecast anticipates future customer requests. A parallel process uses previous routing results to forecast vehicle capacities per area and time window. This parallelization can increase the speed of planning and reduce the impact of flawed demand forecasts. Subsequently, a complex subroutine including revenue management and routing algorithms incrementally adapts capacities and availability controls: Each accepted order provides new information causing a rerouting for capacities, which in turn triggers the need for new fulfillment controls. Once all orders are fixed, a final routing defines the most efficient delivery process to implement.

Of course, alternative processes are conceivable. For instance, a small adaption could be an initial routing for capacities instead of forecasting. Furthermore, alternative models and methods are worth being considered in the implementation. For example, while frequent updates on order acceptance can align revenue and cost considerations, they require more efficient solution methods to remain feasible. Moreover, modelling dependent choice of time windows would improve the understanding of customer behavior. This can be leveraged in the availability control: Shifting low value orders to less popular time windows can improve capacity utilization and maximize revenue.

To benchmark the potential of iterative and integrated planning approaches, both an application-oriented catalogue of problem instances and a complex testing environment are necessary. To serve this purpose, we outline a virtual laboratory in the following section.

4 A Virtual Laboratory for E-Fulfillment Planning

Selecting the best suited planning approach requires to examine the impact of different degrees of integration using a data-driven procedure. In an empirical case study, results can be affected by uncontrollable or unobservable factors. However, a simulation system can serve as a virtual laboratory to conduct computational studies under realistically complex settings. By implementing a library of novel and estab-

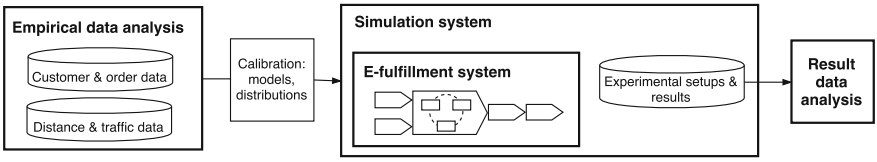


Fig. 2 Structure of the virtual laboratory for e-fulfillment planning

lished approaches, such a laboratory allows us to evaluate and benchmark alternative solutions. Additionally, the included simulation system enables stochastic micro-level sensitivity analyses over the impact of influence factors on all performance areas. Thereby, we can compare different scenarios entirely based on parameterized settings and methods. Appropriate data management ensures the reproducibility of results.

Figure 2 illustrates the high-level structure of the virtual laboratory. The idea is to embed an e-fulfillment system in a framework of complex traffic and demand behavior. As such, the e-fulfillment system is conceptually encapsulated and independent; it could readily be extracted for a stand-alone implementation in an empirical case study.

Any conceivable e-fulfillment system implements a specific process from order capture and delivery planning to recording of delivery data, with different degrees of iterative to fully integrated approaches. To ensure adaptability and extensibility, the system understands a process as a flexible orchestration of multiple, reusable methods. For instance, in the iterative variant of Cleophas and Ehmke [5], a routing method is applied both in the initial capacity estimation and in the final step. We treat routing as an encapsulated module that can be invoked as a service from any step of the process and provides different algorithms. This service is defined by its input and output parameters. Such a service-oriented design avoids redundancies within and between alternative process implementations and supports maintainability.

To allow for realistic experimental settings, real-world data has to be analyzed to calibrate the system. In particular, we will rely on historical data provided by a major e-grocery provider based in Germany. The included transactional data provides insights on customer demand such as basket value distributions and time window choice.

Since travel times in metropolitan areas vary for different daytimes, we consider the time-dependent variant of the VRPTW [6, 7]. To determine time-dependent travel times between customers, we want to include real traffic data from online map providers.

The resulting application-oriented experimental settings ensure the practical relevance of obtained insights. In addition, they can serve as starting point for alternate constraints and sensitivity analyses. For instance, we can observe effects of different degrees of heterogeneity in order basket value.

5 Conclusion

This contribution pointed out the potential of integrated methods for order capture and delivery. Furthermore, it outlined a virtual laboratory that could benchmark such approaches. Future research should additionally improve planning models and methods within the two research areas and consider different optimization objectives. For instance, next to short-term profitability, an e-fulfillment process could account for customer equity and market visibility or consider the reliability and sustainability of delivery strategies.

Acknowledgements This research was supported by a grant from the German Research Foundation (DFG, Grant No. CL605/2-1 and EH449/1-1).

References

1. Agatz, N., Campbell, A., Fleischmann, M., Savelsbergh, M.: Time slot management in attended home delivery. Transp. Sci. **45**(3), 435–449 (2011)
2. Beeson, M., Gill, M., Evans, P., O'Grady, M., Causey, A.: European online retail forecast: 2013 to 2018—European online retail sales continue their double-digit growth. Forrester Res. https://www.forrester.com/European+Online+Retail+Forecast+2013+To+2018/fulltext/-/E-RES115752 (2013). Accessed 7 July 2016
3. Campbell, A.M., Savelsbergh, M.: Decision support for consumer direct grocery initiatives. Transp. Sci. **39**(3), 313–327 (2005)
4. Campbell, A.M., Savelsbergh, M.: Incentive schemes for attended home delivery services. Transp. Sci. **40**(3), 327–341 (2006)
5. Cleophas, C., Ehmke, J.F.: When are deliveries profitable? Bus. Inf. Syst. Eng. **6**(3), 153–163 (2014)
6. Ehmke, J.F., Campbell, A.M.: Customer acceptance mechanisms for home deliveries in metropolitan areas. Eur. J. Oper. Res. **233**(1), 193–207 (2014)
7. Ehmke, J.F., Steinert, A., Mattfeld, D.C.: Advanced routing for city logistics service providers based on time-dependent travel times. J. Comput. Sci. **3**(4), 193–205 (2012)
8. Klein, R., Neugebauer, M., Ratkovitch, D., Steinhardt, C.: Differentiated time slot pricing under routing considerations in attended home delivery. In: SSRN. http://papers.ssrn.com/sol3/papers.cfm?abstract_id=2674061 (2016). Accessed 7 July 2016
9. Quante, R., Meyr, H., Fleischmann, M.: Revenue management and demand fulfillment: matching applications, models, and software. OR Spectr. **31**(1), 31–62 (2009)
10. Yang, X., Strauss, A.K., Currie, C.S., Eglese, R.: Choice-based demand management and vehicle routing in e-fulfillment. Transp. Sci. **50**(2), 473–488 (2014)

Cruise Line Revenue Management: Overview and Research Opportunities

Daniel Sturm and Kathrin Fischer

Abstract While cruise lines share a variety of characteristics with airlines, hotels and especially casinos, revenue management approaches which are suitable for other sectors of the hospitality industry have to be adapted in order to meet the special requirements of cruise lines. Until now, decision support systems, optimization models, as well as pricing and allocation policies for cruise line revenue management have attracted little attention in the field of operations research. Based on a review of the existing literature, research gaps are identified and possible approaches to close these gaps are suggested.

1 Introduction

The cruise industry constitutes an important branch of the global hospitality industry with a worldwide revenue of $39.6 billion and 22.2 million passengers in 2015 [5]. Hence, this sector of the hospitality industry is of significant economic relevance and optimization approaches for cruise lines can have a considerable financial impact. Due to exceptionally high occupancy rates [23], the cruise industry is especially well suited for revenue management (RM), which can improve the allocation of the scarce cruise fleet resources to the most valuable booking requests.

Nevertheless, cruise line revenue management (CLRM) has not yet been given much attention by the OR community. Accordingly, a recent review of CLRM [21] focusses on marketing without capturing OR perspectives. Only few RM approaches tailored to the specific challenges in the cruise industry have yet been suggested. Therefore, the contributions of this work are a succinct description of the distinctive characteristics of CLRM, a systematic analysis of the existing literature and suggestions for the integration of the specific CLRM characteristics into RM approaches.

D. Sturm (✉) · K. Fischer
Institute for Operations Research and Information Systems,
Hamburg University of Technology, Am Schwarzenberg-Campus 4,
21073 Hamburg, Germany
e-mail: daniel.sturm@tuhh.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_59

2 Cruise Line Revenue Management

The problem of CLRM is associated with the general problem of RM, i.e. when to sell which amount of a perishable resource to which customer at which price in order to maximize revenue. On a cruise ship, the perishable resource consists of cabins, which become worthless with the ship's departure. Directly related problems arise in traditional applications of RM (i.e. airlines and hotels), as well as in a variety of other service industries. A comprehensive overview on RM in general is offered by [3], while [8] review RM approaches for the hospitality industry.

In order to apply RM, well-known prerequisites have to hold, namely fixed and perishable capacities, the possibility of market segmentation, advance bookings, comparably low marginal costs and fluctuating demand. All of these are fulfilled by the cruise line industry [10]. Based on [2, 14, 15, 17, 19, 22, 23] six additional, distinguishing characteristics of CLRM can be identified which are relevant for RM approaches from an OR perspective:

Multiple Capacity Limitations While for a flight leg or a hotel stay only a single capacity limit has to be considered (i.e. seats or rooms), the number of passengers on a cruise is limited by the quantity of cabins of different categories, as well as by the number of lifeboat seats. Moreover, passengers often book additional "air packages", i.e. corresponding transfer flights, which can also have a limited availability. Further limitations can be imposed by on-board facilities or shore excursions.

Guest Pricing Usually, in hotels each room category receives an individual price tag. In contrast, cruise lines price their guests individually. The base fare for a cabin usually consists of a "double occupancy" (i.e. the price for two adult passengers), and each additional passenger is additionally billed.

Customer Value A major and growing part of cruise revenue stems from the on-board expenses of passengers (e.g. food, beverages, entertainment, sports or shore excursions). This implies that even a notably discounted fare may be overcompensated by increased on-board revenue due to additionally attracted passengers. Moreover, as cruise passengers return often to cruising in general [4], customer satisfaction is paramount in order to preserve customer loyalty to one's own cruise line.

Demand Substitution A cruise ship, unlike a hotel, offers a multitude of different cabin types, each with distinctive attributes. These types can be summarized into few cabin type groups, which can be used for market segmentation. This also implies that customers are, to a certain degree, indifferent between the cabin types of one group. But, as types within the same group can still exhibit considerable physical differences, demand substitution between different cabin types can occur.

Overbooking Airlines and hotels use overbooking to compensate for cancellations and no-shows. As cruise lines impose strict cancellation policies upon their customers, cancellations shortly before and no-shows at departure are uncommon. Moreover, compensation of refused passengers is difficult: Unlike with hotels, a cruise passenger cannot be "walked" to a comparable cruise in the vicinity.

Customer Choice Behaviour Booking a cruise is a dynamic and highly interactive process, which is almost entirely conducted and controlled by travel agents. They receive customer requests, offer reservation options for multiple cruises, finalize bookings, collect deposits and handle overbooked cruises. At each stage of this elaborate process, customers can be offered a variety of customized incentives (e.g. discounts) in order to influence their choice behaviour and to control demand.

Despite the aforementioned particularities, similarities to other industries do exist. For example, in container shipping multiple capacity limitations hold as well as the number of container slots as well as the ship's maximum loading weight limit the number of loadable containers (see, e.g., [24]). Also, a resemblance to casino hotels can be recognized, especially with respect to the importance of individual customer value due to on-site spending (i.e. gambling, retail sales, food and beverage consumption) and customer loyalty considerations (see, e.g., [2, 9]).

As mentioned in Sect. 1, not much attention has yet been given to the theory and practice of CLRM [2, 8, 17, 21], especially not by OR researchers. An analysis of the existing literature on CLRM is summarized in Table 1. Besides a short description of the methodological approach, a categorization with respect to the consideration of the distinguishing characteristics of CLRM as presented above is shown.

Table 1 Consideration of distinguishing CLRM characteristics in OR based RM approaches

Reference	Problem, objective, method	MC	GP	CV	DS	OB	CC
Ladany and Arbel [11] *1991*	Determination of a profit maximizing price differentiation strategy (i.e. market segment sizes and respective fares)	∘	∘	∘	∘	∘	∘
Biehn [2] *2006*	Cabin capacity allocation maximizing revenue, LP with deterministic demand	•	•	∘	∘	∘	∘
Ji and Mazzarella [10] *2007*	Booking limit determination using nested and dynamic class allocation	∘	∘	∘	∘	∘	∘
Li [12] *2010*	Cabin capacity allocation maximizing revenue using stochastic ILP and DP	•	•	∘	∘	∘	∘
Maddah et al. [17] *2010*	Cabin capacity control maximizing revenue using DP and heuristics while considering stochastic demand	•	•	∘	∘	∘	∘
Ma and Sun [16] *2012*	Cabin capacity control maximizing revenue using nested and improved class allocation using Bayesian inference	∘	∘	∘	∘	∘	∘
Li [13] *2014*	Optimal overbooking level determination using real options valuation	∘	∘	∘	∘	•	∘
Li et al. [14] *2014*	Cabin capacity allocation maximizing revenue using deterministic ILP	•	•	•	∘	∘	∘

MC Multiple Capacity Limitations, *GP* Guest Pricing, *CV* Customer Value
DS Demand Substitution, *OB* Overbooking, *CC* Customer Choice Behaviour

As can be concluded from Table 1, the literature on CLRM is not only scarce, but also heterogeneous with respect to the considered problem formulations, methodologies, and especially the consideration of the specific characteristics of the cruise industry identified above. As the unique and important characteristics of CLRM have not been taken appropriately into account so far in the literature, there are significant gaps in the research on CLRM from an OR perspective. Hence, this area offers promising research opportunities for developing new RM approaches, some of which are discussed below.

3 Research Opportunities

The possible approaches to close existing research gaps in CLRM suggested below affect all subtasks of RM, namely market segmentation, forecasting, dynamic pricing, as well as capacity and overbooking control.

An improved market segmentation and a reduced risk of demand substitution can be achieved by employment of effective segmentation criteria which do not solely rely on physical differences between cabin types. Even though this issue is more related to consumer marketing research than to OR, effectively applied segmentation criteria and their integration into pricing, capacity and overbooking control will improve the overall results of the respective RM approach.

Regarding capacity control and dynamic pricing, all relevant capacity limitations have to be incorporated into optimization models and heuristics used for capacity allocation (i.e. determination of booking limits or bid prices) and fare determination by formulating and incorporating the associated mathematical constraints into these models. As cruise lines often procure a quota on a variety of airways for use in transfer flights, these capacities should also be considered as limiting when deciding on fares or the acceptance of potential booking requests with “air packages”. A related problem has been tackled by [15], who determine the least-cost assignment of cruise passengers to a predetermined set of available transfer flights. Also, group bookings could be considered as it has been done for hotels, e.g. by [7].

Moreover, the consecutive stages of the cruise booking process should be incorporated into dynamic capacity control and pricing models, in order to capture the high degree of interactivity between cruise lines (or travel agents) and prospective customers. Especially individual customer choice behaviour has to be considered in order to mitigate the risk of demand substitution. RM models could determine at which stage of the booking process an incentive of a certain kind (e.g. fare discount, upgrades) should be offered to a prospective customer. Thus, by inducing a favourable customer choice behaviour (e.g. upgrading, purchase of packages), existing demand can be shaped optimally. This approach could be extended to include multiple cruises, i.e. to use incentives to shift demand between similar cruises in temporal proximity by optimizing not only a singular cruise, but a set of cruises.

Furthermore, the individual customer value should be properly parametrized and incorporated into capacity allocation and pricing optimization models and heuristics [18]. This also includes customer loyalty, which on the one hand, when sufficiently high, leads to a returning customer and hence the generation of future additional revenue and on the other hand could lead to the acquisition of new customers due to recommendations. Combining monetary value and loyalty, the resulting passenger's customer lifetime value might pose another important decision criterion with respect to capacities, fares and incentives offered to him. An example of customer lifetime value determination for a cruise ship company is offered by [1].

According to [23], travel agents usually offer customers reservation options on different cruises throughout the comparatively long booking horizon (up to one year before departure), letting them delay their final booking decision. Overbooking of cruise fleet capacities is common in this phase of the booking process as the cancellation of a high number of reservation options is to be expected. Thus, this circumstance should be explicitly integrated into overbooking models, which in turn are to be combined with capacity control and pricing models.

One important implication is inherent to the aforementioned suggestions: Instead of demand aggregated on cabin type level, individual booking requests with their constitutive attributes (e.g. party mix, cabin type requested, customer lifetime value, individual choice behaviour) have to be considered in all steps of the RM process. This will, beyond doubt, present additional challenges with respect to demand forecasting, as not only aggregated demand quantities, but the specific composition of expected booking requests will have to be predicted. A starting point with regard to forecasting can be found in [20], who compare a variety of forecasting models with respect to their applicability in the cruise industry. However, they do not focus on individual booking requests, but on aggregate demand for a single cabin category.

Finally, alternative objective functions and combinations thereof should be considered, like occupancy rate maximization, customer satisfaction maximization by minimization of the deviation of customer requests (e.g. with regard to cabin types or departure dates) from actual bookings or fair treatment of passengers with respect to discounts and other beneficial treatments.

4 Conclusion

The presently available OR based CLRM research is scarce and heterogeneous. Most approaches consider only few of the special characteristics of CLRM, and no comprehensive RM methodology taking into account all or most of the relevant features has been developed yet. Existing optimization models and heuristics should therefore be extended, integrating as many of the aspects discussed in Sect. 3 as possible, in order to subsequently develop customized optimal pricing and capacity control policies. As the effects and implications of new RM approaches should be evaluated by employing simulations [6], another important step consists of the design of a customized simulation environment for CLRM with which the developed approaches

can be tested. Hence, from an OR perspective, CLRM constitutes a promising area for future research in the field of RM.

References

1. Berger, P.D., Weinberg, B., Hanna, R.C.: Customer lifetime value determination and strategic implications for a cruise-ship company. J. Database Mark. Cust. Strategy Manage. **11**(1), 40–52 (2003)
2. Biehn, N.: A cruise ship is not a floating hotel. J. Revenue Pricing Manage. **5**(2), 135–142 (2006)
3. Chiang, W.-C., Chen, J.C., Xu, X.: An overview of research on revenue management: current issues and future research. Int. J. Revenue Manage. **1**(1), 97–128 (2007)
4. Cruise Lines International Association (CLIA): 2014 North American Cruise Market Profile. http://www.cruising.org/docs/default-source/research/clia_naconsumerprofile_2014.pdf (2015). Accessed 8 Aug 2016
5. Cruise Market Watch: 2015 World Wide Market Share. http://www.cruisemarketwatch.com/market-share/ (2016). Accessed 8 Aug 2016
6. Frank, M., Friedemann, M., Schröder, A.: Principles for simulations in revenue management. J. Revenue Pricing Manage. **7**(1), 7–16 (2008)
7. Guadix, J., Cortés, P., Onieva, L., Muñuzuri, J.: Technology revenue management system for customer groups in hotels. J. Bus. Res. **63**(5), 519–527 (2010)
8. Guillet, B.D., Mohammed, I.: Revenue management research in hospitality and tourism. A critical review of current literature and suggestions for future research. Int. J. Contemp. Hosp. Manage. **27**(4), 526–560 (2015)
9. Hendler, R., Hendler, F.: Revenue management in fabulous Las Vegas: combining customer relationship management and revenue management to maximise profitability. J. Revenue Pricing Manage. **3**(1), 73–79 (2004)
10. Ji, L., Mazzarella, J.: Application of modified nested and dynamic class allocation models for cruise line revenue management. J. Revenue Pricing Manage. **6**(1), 19–32 (2007)
11. Ladany, S.P., Arbel, A.: Optimal cruise-liner passenger cabin pricing policy. Eur. J. Oper. Res. **55**(2), 136–147 (1991)
12. Li, B.: Modelling for cruise two-dimensional online revenue management system. Int. J. Digit. Content Appl. **4**(6), 72–78 (2010)
13. Li, B.: A cruise line dynamic overbooking model with multiple cabin types from the view of real options. Cornell Hosp. Q. **55**(2), 197–209 (2014)
14. Li, Y., Miao, Q., Wang, B.X.: Modeling a cruise line revenue management problem. J. Revenue Pricing Manage. **13**(3), 247–260 (2014)
15. Lieberman, W.H., Dieck, T.: Expanding the revenue management frontier: optimal air planning in the cruise industry. J. Revenue Pricing Manage. **1**(1), 7–18 (2002)
16. Ma, D., Sun, J.: Revenue management system for the cruise industry: a simulation study. In: Papathanassis, A., Breitner, M.H., Schoen, C., Guhr, N. (eds.) Cruise Management. Information and Decision Support Systems, pp. 223–232. Gabler, Wiesbaden (2012)
17. Maddah, B., Moussawi-Haidar, L., El-Taha, M., Rida, H.: Dynamic cruise ship revenue management. Eur. J. Oper. Res. **207**(1), 445–455 (2010)
18. von Martens, T., Hilbert, A.: Customer-value-based revenue management. J. Revenue Pricing Manage. **10**(1), 87–98 (2011)
19. Phillips, R.L.: Pricing and Revenue Optimization. Stanford University Press, Stanford (2005)
20. Sun, X., Gauri, D.K., Webster, S.: Forecasting for cruise line revenue management. J. Revenue Pricing Manage. **10**(4), 306–324 (2011)
21. Sun, X., Jiao, Y., Tian, P.: Marketing research and revenue optimization for the cruise industry: a concise review. Int. J. Hosp. Manage. **30**(3), 746–755 (2011)

22. Talluri, K.T., van Ryzin, G.J.: The Theory and Practice of Revenue Management. Springer, New York (2004)
23. Toh, R.S., Rivers, M.J., Ling, T.W.: Room occupancies: cruise lines out-do the hotels. Int. J. Hosp. Manage. **24**(1), 121–135 (2005)
24. Zurheide, S., Fischer, K.: Revenue management methods for the liner shipping industry. Flex. Serv. Manuf. J. **27**(2), 200–223 (2015)

Part XV
Production and Operations Management

Regionalized Assortment Planning for Multiple Chain Stores

Hans Corsten, Michael Hopf, Benedikt Kasper and Clemens Thielen

Abstract In retail, assortment planning refers to selecting a subset of products to offer that maximizes profit. Assortments can be planned for a single store or a retailer with multiple chain stores where demand varies between stores. In this paper, we assume that a retailer with a multitude of stores wants to specify her offered assortment. To suit all local preferences, regionalization and store-level assortment optimization are widely used in practice and lead to competitive advantages. When selecting regionalized assortments, a trade-off between expensive, customized assortments in every store and inexpensive, identical assortments in all stores that neglect demand variation is preferable. We formulate a stylized model for the regionalized assortment planning problem (APP) with capacity constraints and given demand. In our approach, a common assortment that is supplemented by regionalized products is selected. While products in the common assortment are offered in all stores, products in the local assortments are customized and vary from store to store. Concerning the computational complexity, we show that the APP is strongly NP-hard. The core of this hardness result lies in the selection of the common assortment. We formulate the APP as an integer program and provide algorithms and methods for obtaining approximate solutions and solving large-scale instances. Lastly, we perform computational experiments to analyze the benefits of regionalized assortment planning depending on the variation in customer demands between stores.

M. Hopf (✉) · C. Thielen
Department of Mathematics, University of Kaiserslautern, Paul-Ehrlich-Str. 14, 67663 Kaiserslautern, Germany
e-mail: hopf@mathematik.uni-kl.de

C. Thielen
e-mail: thielen@mathematik.uni-kl.de

H. Corsten · B. Kasper
Department of Business Studies and Economics, University of Kaiserslautern, Gottlieb-Daimler-Str. 42, 67663 Kaiserslautern, Germany
e-mail: corsten@wiwi.uni-kl.de

B. Kasper
e-mail: benedikt.kasper@wiwi.uni-kl.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_60

1 Introduction

A retailer with a multitude of stores has two basic strategies for specifying her offered assortment:

- Every store has a customized assortment. Here, all demand differences can be considered, but customized assortments are expensive to maintain.
- Assortments in all stores are the same. Here, the assortments may not be optimized to suit all local preferences, but with a single assortment, economies of scale and a recognition value can be generated.

In this paper, we analyze the benefit of mixed strategies, i.e., the selection of a *common assortment* that is supplemented by regionalized products. Thus, products in the common assortment are offered in all stores, while products in the local assortments are customized and vary from store to store.

The assortment planning problem (APP) is considered in both operations research and retail literature in various settings. For extensive reviews, see [6, 7]. Generally, most researchers take shelf space [1], inventory [4], or pricing decisions [9] into account. Regionalization and store-level assortment optimization lead to competitive advantages and are widely used in practice [3, 5]. In [6], the authors describe this as follows: Chain store management dictates a portion of the assortment that is carried in all stores, while the remainder is chosen to satisfy local customer preferences. Surprisingly, very little research is done in this context. In [3, 8], the authors take a step in that direction, but in their models, a recognition value and economies of scale cannot be generated.

We propose an alternative solution method that reflects industry practice. Items for the common assortment and items for the local assortments are selected simultaneously in order to maximize the total profit. We show that this problem is strongly NP-hard and present a heuristic that is able to tackle large-scale instances that cannot be solved within a reasonable amount of time by applying a commercial solver to a standard integer programming formulation. Moreover, we evaluate the quality of our algorithm in several computational experiments.

2 Formulation of the APP as an Integer Program

In this section, we formulate a stylized model for the APP that is obtained by simplifying the original problem using some reasonable assumptions.

Assumption 1 We develop our model using the following assumptions:

- The assortment consists of standardized products, offered at standardized shelf space (i.e., all products have unit size).
- Every store has the same capacity.
- Demand can be estimated for every product and every store.

- Products that are assigned to the common assortment are offered in every store, while products in a local assortment are offered only in this particular store.

The most restrictive assumption is that all products have unit size. However, we can view the unit size as a standardized area (e.g., 1 m^2) that is occupied by each item. Then, an item corresponds to the number of units of a particular product that fit into this area (e.g., 1000 pencils, 4 toasters). The assumption that every store has the same capacity can be loosened (see Sect. 3).

Now, we describe the APP as a variant of the multiple knapsack problem. We are given m bins (stores) with size K (capacity of the store) in which we want to pack unit size items (products). In total, there are n different items. With each item j, we associate $m+1$ different profits $w_j \geq 0$ and $v_{jk} \geq 0$ for $k = 1, \ldots, m$. We obtain profit w_j if the item is packed into all bins (i.e., packed into the common assortment) and profit v_{jk} if item j is packed into bin k, but there is at least one bin in which we do not pack it. Using this notation, we can model the problem as the following integer program:

$$
\begin{aligned}
& \textbf{maximize} && \sum_{j=1}^{n} w_j x_j + \sum_{j=1}^{n} \sum_{k=1}^{m} v_{jk} y_{jk} && \\
\text{(APP)} \quad & \textbf{subject to} && \sum_{j=1}^{n} (x_j + y_{jk}) \leq K && \forall k \in \{1, \ldots, m\} \\
& && x_j + y_{jk} \leq 1 && \forall j \in \{1, \ldots, n\}, k \in \{1, \ldots, m\} \\
& && x_j, y_{jk} \in \{0, 1\} && \forall j \in \{1, \ldots, n\}, k \in \{1, \ldots, m\}
\end{aligned}
$$

where

$$
x_j = \begin{cases} 1, & \text{if item } j \text{ is packed into the common assortment} \\ 0, & \text{else} \end{cases}
$$

and

$$
y_{jk} = \begin{cases} 1, & \text{if item } j \text{ is packed into bin k (but not into the common assortment)} \\ 0, & \text{else} \end{cases}
$$

The profits w_j and v_{jk} are composed of estimated revenue and cost as follows: Let r_{jk} be the estimated revenue of product j in store k, c_{djk} the cost of type d (e.g., procurement, transportation, storage cost) of product j in store k, c^x the assignment cost for assigning a product to the common assortment, and c^y_k the assignment cost for assigning a product to the local assortment of store k. Then, we can write the objective function of the APP as

$$
\sum_j \sum_k r_{jk}(x_j + y_{jk}) - \sum_j \sum_k \sum_d c_{djk}(x_j + y_{jk}) - \sum_j \sum_k (c^x x_j + c^y_k y_{jk}).
$$

where

$$w_j = \sum_k \left(r_{jk} - \sum_d c_{djk} - c^x \right) \quad \text{and} \quad v_{jk} = r_{jk} - \sum_d c_{djk} - c^y_k.$$

Thus, the estimation of the parameters w_j and v_{jk} is essentially the estimation of the revenues and costs of the products.

The main goal of this model for the APP is to decide which products to place in the common assortment and which in the local assortments, given estimated parameters w_j and v_{jk}.

3 Computational Complexity and Algorithmic Approach

In this section, we analyze the computational complexity of the APP. Theorem 2 states that the problem is strongly NP-hard. Thus, unless P = NP, there is no algorithm that solves the APP exactly in polynomial time.

Theorem 2 *The APP is strongly NP-hard.*

The proof uses a reduction from the satisfiability problem SAT and can be found in [2].

As it turns out, commercial solvers are unable to obtain optimal solutions within a reasonable amount of time even for instances with 300 stores and 15.000 products. This motivates the development of algorithms that run fast and produce close to optimal solutions. We now present a greedy heuristic that is used in Sect. 4 to solve large-scale instances of the APP. The idea of the algorithm is to first neglect the advantages of using the common assortment and start with the best local assortment for each store (independent of the others). Then, in each step, the algorithm adds the item that currently grants the largest gain in total profit to the common assortment. Denoting the current common assortment by C and the current set of items in bin k by I_k, the algorithm can be formulated as follows:

Algorithm 1

1: Let $C := \emptyset$ and, for each bin k, let I_k be the set containing the K items j with the highest values v_{jk}.

2: For $j \notin C$, let $u_j := w_j - \left(\sum_{k: j \notin I_k} \min_{l \in I_k} v_{lk} \right) - \sum_{k: j \in I_k} v_{jk}$.

3: If $u_j \leq 0$ for all j or $|C| = K$, stop; else add the item $j \notin C$ with the largest value u_j to the common assortment C, update the sets I_k by removing j from I_k if it is contained in I_k and removing an item j' with minimum value $v_{j'k}$ from I_k otherwise, and go to Step 2.

Each time we update I_k, its size reduces by one since we remove one item and put it into C. Observe that Algorithm 1 also works in the case where the capacities of the

bins differ. Then, in the first step, for each bin k, we pack the best K_k items, where K_k denotes the individual capacity of bin k. Moreover, the second stopping criterion changes to $|C| = \min_k K_k$. The running time of Algorithm 1 is in $\mathcal{O}(nm(\log(n) + K))$.

Additionally, a preprocessing strategy can be used in order to identify items that will never be contained in the common assortment. It can easily be seen that choosing the better one of the best packings with $|C| = K$ (ALG_2) and $C = \emptyset$ (ALG_3) yields an approximation ratio of 2. If we modify Algorithm 1 (ALG_1) slightly so that it compares the computed solution with the one obtained by ALG_2 and chooses the better one, it also obtains this approximation guarantee. More detailed results on approximation guarantees can be found in [2].

4 Experimental Results

In this section, we present computational experiments in order to compare the solution quality obtained by ALG_1, ALG_2, and ALG_3. In particular, we are interested in how large the common assortment profits w_j must be in relation to the local profits v_{jk} in order to see substantial benefits from mixing common and local assortments as in ALG_1 (when compared to using only the common assortment as in ALG_2 or only the local assortments as in ALG_3).

We randomly generate the values v_{jk} and w_j. However, it seems to be a reasonable assumption that the values v_{jk} are dependent for a fixed item j although the cost of providing item j (e.g., the transportation cost) might vary for different stores. Therefore, we consider three scenarios where we draw values v_j uniformly at random from $[0, 1]$ independently for all j and then

- set $v_{jk} := v_j$ for all k (*total dependence*), or
- draw r_k uniformly from $[-0.5p, 0.5p]$ and set $v_{jk} := \max(0, v_j + r_k)$, where p is a model parameter (*intermediate dependence*), or
- draw all values v_{jk} uniformly and independently from $[0, 1]$ (*total independence*).

In order to generate the values w_j, we draw values q_j uniformly at random from $[0.95, 1.05]$ and set $w_j := q_j b \sum_k v_{jk}$, where b represents the financial gains when a product is in the common assortment (e.g., from economies of scale or recognition value). We consider 100 equidistant values of b in $[1, 2]$ and generate 100 instances for each of these values and four different settings concerning the dependence of the values v_{jk} (total independence, intermediate dependence with $p = 0.75$ and $p = 0.95$, and total independence). The optimal profit for each instance is calculated by solving the IP formulation using Gurobi 6.5.

We observe that, when b is too small or too large, algorithms ALG_2 and/or ALG_3 already yield close to optimal solutions. Therefore, for each of the four settings concerning dependence of the values v_{jk}, we concentrate on three values of b that are neither too small nor too large. In the setting of total independence, we consider $b \in \{1.2, 1.35, 1.5\}$, for $p = 0.75$ and total dependence, we consider

Table 1 Average ratios $\frac{\text{OPT}}{\text{ALG}_i}$ for $i = 2, 3$ (for small instances) and $\frac{\text{ALG}_1}{\text{ALG}_i}$ (for large instances)

Value b	Small	Medium	Large	Small	Medium	Large
Instance size	Small	Small	Small	Large	Large	Large
Dependence						
Total dependence	1.01|1.02	1.00|1.05	1.00|1.09	1.01|1.02	1.00|1.05	1.00|1.09
$p = 0.75$	1.06|1.01	1.04|1.03	1.03|1.06	1.06|1.01	1.04|1.03	1.03|1.06
$p = 0.95$	1.07|1.02	1.05|1.05	1.04|1.08	1.08|1.02	1.05|1.05	1.04|1.08
Total independence	1.18|1.01	1.09|1.05	1.04|1.11	1.20|1.01	1.10|1.03	1.05|1.09

$b \in \{1.01, 1.05, 1.09\}$, and for $p = 0.95$, we consider $b \in \{1.04, 1.09, 1.14\}$. Moreover, we consider two different instance sizes (small and large), where $(n, m, K) = (1500, 50, 750)$ and $(n, m, K) = (50.000, 150, 25.000)$, respectively. For all considered instances, ALG_1 obtains nearly optimal solutions (i.e., the average ratio $\frac{\text{OPT}}{\text{ALG}_1}$ of the profits of an optimal solution and the algorithm is below 1.01). Table 1 shows the average ratios $\frac{\text{OPT}}{\text{ALG}_i}$ for $i = 2, 3$ obtained by ALG_2 and ALG_3.[1] Here, we observe that the gain in profit from using an optimized assortment compared to one of the solutions produced by ALG_2 or ALG_3 can be up to 20%.

5 Conclusion

We have formulated a stylized model for the regionalized assortment planning problem (APP) and have shown that solving the APP can lead to significant profit gains for a retailer with multiple chain stores. Moreover, we proposed a local improvement heuristic that computes close to optimal solutions in polynomial time. In a next step, we will test this algorithm with real world data. For future research, we propose extensions of the model such as sub-regions (here, we also obtain a profit gain when a product is placed in a certain set of stores) or individual item sizes.

References

1. Chen, M.-C., Lin, C.-P.: A data mining approach to product assortment and shelf space allocation. Expert Syst. Appl. **32**, 979–986 (2007)
2. Corsten, H., Hopf, M., Kasper, B., Thielen, C.: Regionalized assortment planning for multiple chain stores: complexity, approximability, and solution methods. Report in Wirtschaftsmathematik, vol. 162. University of Kaiserslautern (2016)

[1] Since the large instances could not be solved to optimality within a reasonable amount of time by using Gurobi, we provide the ratios $\frac{\text{ALG}_1}{\text{ALG}_i}$ instead for these instances.

3. Fisher, M., Vaidyanathan, R.: A demand estimation procedure for retail assortment optimization with results from implementations. Manage. Sci. **60**, 2401–2415 (2014)
4. Honhon, D., Gaur, V., Seshadri, S.: Assortment planning and inventory decisions under stockout-based substitution. Oper. Res. **58**, 1364–1379 (2010)
5. Hwang, M., Bronnenberg, B.J., Thomadsen, R.: An empirical analysis of assortment similarities across US supermarkets. Market. Sci. **29**, 858–879 (2010)
6. Kök, A.G., Fisher, M.L., Vaidyanathan, R.: Assortment planning: review of literature and industry practice. In: Agrawal, N., Smith, S.A. (eds.) Retail Supply Chain Management vol. 223, pp. 99–153. Springer, US (2009)
7. Mantrala, M.K., Levy, M., Kahn, B.E., Fox, E.J., Gaidarev, P., Dankworth, B., Shah, D.: Why is assortment planning so difficult for retailers? A framework and research agenda. J. Retail. **85**, 71–83 (2009)
8. Rooderkerk, R.P., van Heerde, H.J., Bijmolt, T.H.A.: Optimizing retail assortments. Market. Sci. **32**, 699–715 (2013)
9. Wang, R.: Capacitated assortment and price optimization under the multinomial logit model. Oper. Res. Lett. **40**, 492–497 (2012)

Optimizing Machine Spare Parts Inventory Using Condition Monitoring Data

Sonja Dreyer, Jens Passlick, Daniel Olivotti, Benedikt Lebek and Michael H. Breitner

Abstract In the manufacturing industry, storing spare parts means capital commitment. The optimization of spare parts inventory is a real issue in the field and a precise forecast of the necessary spare parts is a major challenge. The complexity of determining the optimal number of spare parts increases when using the same type of component in different machines. To find the optimal number of spare parts, the right balance between provision costs and risk of machine downtimes has to be found. Several factors are influencing the optimum quantity of stored spare parts including the failure probability, provision costs and the number of installed components. Therefore, an optimization model addressing these requirements is developed. Determining the failure probability of a component or an entire machine is a key aspect when optimizing the spare parts inventory. Condition monitoring leads to a better assessment of the components failure probability. This results in a more precise forecast of the optimum spare parts inventory according to the actual condition of the respective component. Therefore, data from condition monitoring processes are considered when determining the optimal number of spare parts.

S. Dreyer (✉) · J. Passlick · D. Olivotti · M.H. Breitner
Information Systems Institute, Leibniz Universität Hannover,
Königsworther Platz 1, 30167 Hannover, Germany
e-mail: dreyer@iwi.uni-hannover.de

J. Passlick
e-mail: passlick@iwi.uni-hannover.de

D. Olivotti
e-mail: olivotti@iwi.uni-hannover.de

M.H. Breitner
e-mail: breitner@iwi.uni-hannover.de

B. Lebek
bhn Dienstleistungs GmbH & Co. KG, Hans–Lenze–Straße 1, 31855 Aerzen, Germany
e-mail: lebek.benedikt@bhn-services.com

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_61

1 Introduction

Optimizing machine spare parts inventory means finding the right balance between spare parts costs and costs caused by machine downtimes [6]. Therefore, inventory control is an important topic in operations management [1]. The presented model minimizes the costs by determining the optimal number of available spare parts. The costs are optimized for a type of component which is installed multiple times in the production site. It is possibles to reduce the amount of available spare parts because one spare part can be used in several machines. To find the optimal number of spare parts, the probability of default of each component has to be determined. In the presented optimization model both the current state of the respective component received through sensor data as well as empirical values are used to predict the probability of default of a component. Further information are considered such as the potential downtime costs, which are compared to each other.

The model is based on a new service concept that makes it possible to adjust the number of available spare parts in each period. A general concept is problematic because when buying a spare part it has to stay in stock until it is needed [2]. In the newly developed service concept the spare parts do not have to be bought but a lump–sum fee for the provision is charged. This lump–sum fee functions as a payment for the provision of a spare part. The advantage is, when needing a spare part because of a component failure, it can be installed directly. When the optimal stock amount decreases, spare parts can be returned. Therefore, it is possible to decide anew in each period how many spare parts should be available to minimize the costs.

2 Optimization Model

The model optimizes the sum of provision costs and the expected downtime costs for one period by determining the optimal number of spare parts. The optimization is performed through a stochastic model in conjunction with an algorithm (Fig. 1). The central assumption is that it is only checked at the end of a period whether a component is defective or not. This leads to the possibility of repairing components with high downtime costs preferentially. Therefore, the installed components are sorted in descending order according to the by the component caused machines downtime costs. A decision tree illustrates all possible combinations of faultless and defective components. Each branch represents a discrete and stochastically independent event, thus describing one possible combination of defective and faultless components, which can be found at the end of a period. Thereby, the first component in the branch represents the component causing the highest downtime costs. It results in a number of branches of $b = 2^c$. The probability of default is influenced by sensor data and empirical values. The sensor data is received through condition monitoring [3] whereas empirical values result from expert knowledge. This data may lead to a Weibull distribution of life expectancy depending on the probability of default [4, 5].

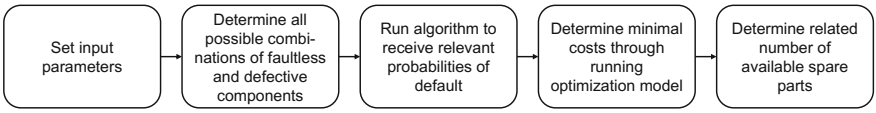


Fig. 1 General procedure to determine the optimal number of available spare parts

Sets:

$i = (1, ..., c)$ considered component of in total c components where $i = 1$ is the component which causes the highest downtime costs
$j = (1, ..., b)$ considered branch of in total b branches
$k = (0, ..., c - 1)$ possible number of available spare parts

Parameters:

Cd_i downtime costs of the machine with the installed component i
Cp provision costs for one spare part
e_i effect on the machine breakdown
p_{ik} probability of downtime costs; *determined by algorithm*
pd_i total probability of default
pe_i probability of default resulting from empirical values
ps_i probability of default resulting from sensor data
w weighting of probability resulting from sensor data
cs_{ij} 0, if component status is faultless, 1 else
q_{ij} probability of component within the branch pd_i if cs_{ij} is 1, $1 - pd_i$ else; *with* pd_i *from* (4)
y_{ijk} 1, if downtime costs have to be paid, 0 else

Decision variable:

x number of available spare parts

$$Minf(x) = \begin{cases} x \times Cp + \sum_{i=x+1}^{c} p_{ix} \times Cd_i \times e_i & \forall x < c \\ x \times Cp & x = c \end{cases} \tag{1}$$

$$0 \leq x \leq c \qquad x \in \mathbb{N}_0 \tag{2}$$

$$0 \leq e_i \leq 1 \qquad \forall i \tag{3}$$

$$pd_i = w \times ps_i + (1 - w) \times pe_i \qquad \forall i \tag{4}$$

$$0 \leq p_{ik}, pd_i, pe_i, ps_i \leq 1 \tag{5}$$

$$0 \leq w \leq 1 \tag{6}$$

$$cs_{ij} \in \{0, 1\} \qquad \forall\, i \,and\, j \tag{7}$$

$$y_{ijk} \in \{0,1\} \qquad \forall\, i,\, j \,and\, k \tag{8}$$

$$b, c \in \mathbb{N}\backslash\{0\} \tag{9}$$

The objective function (1) minimizes the costs consisting of the sum of provision costs and the expected downtime costs. All parameters refer to one period. It is assumed that the number of available spare parts must not exceed the number of installed components and not be less than zero (2). This is because the costs are minimized for one period. The effect of the component on the machine downtime has to be between zero when there is no effect and one in case of a complete breakdown of the machine when the component is defective (3). Sensor data as well as empirical values are considered to determine the total probability of default of a component (4). Constraint (5) ensures that the probabilities are between zero and one. To receive a weighted average of the probabilities, the weighting factor must be between zero when only considering empirical values and one when only sensor data are considered (6). The component status is defined as a binary variable according to the decision tree (7). Depending on whether the downtime costs have to be paid for the considered component the variable in equation (8) is zero or one. This is determined by the developed algorithm. Equation (9) ensures that the number of branches and installed components is a positive integer.

To solve the objective function (1) the respective probability of downtime costs p_{ik} has to be determined for each possible combination of spare parts. Thus, all installed components have to be considered. This is done by an algorithm. To avoid the calculation of trivial cases, in the algorithm it is assumed that $k < i$.

(step 1) Set $i = 1, j = 1$ and $k = 0$.
(step 2) If $cs_{ij} = 0$, set $y_{ijk} = 0$.
(step 3) Else: If $\sum_{a_1=1}^{i} cs_{a_1 j} \leq k$, set $y_{ijk} = 0$.
(step 4) Else set $y_{ijk} = 1$.
(step 5) Increment j by 1. If $j \leq 2^c$, go to (step 2).
(step 6) Else calculate $p_{ik} = \sum_{a_3=1}^{b} (y_{ia_3k} \times \prod_{a_2=1}^{c} q_{a_2a_3})$.
(step 7) Increment i by 1. If $i \leq c$, set $j = 1$ and go to (step 2).
(step 8) Else increment k by 1. If $k < c$, set $i = k + 1$ and $j = 1$ and go to (step 2).
(step 9) Else terminate.

The number of spare parts is set to zero (step 1). Furthermore, (step 1) sets the considered component to the component with the largest potential downtime costs. In the beginning, the first branch is considered. In (step 2), (step 3) and (step 4) it is checked whether the component is defective or faultless. Based on the finding it can be decided if the downtime costs occur for the component in the viewed state. This is done by determining the number of defective components which cause higher machine downtime costs. Afterwards, it is looked at the following branch until the last branch is reached (step 5). The probability of downtime costs is calculated for the viewed component (step 6). The considered component is changed to the next component in the ranking (step 7). The number of the branch is set back to one.

When all relevant components are considered, the number of available spare parts is incremented by one (step 8). The procedure starts anew. The algorithm is executed until the maximum number of available spare parts is reached (step 9). The resulting probabilities of downtime costs p_{ik} make solving the objective function (1) possible.

3 Experimental Results

Based on the presented model optimizations were conducted within a test case using ten installed components of the same type. The configuration of the input parameters is summarized in Table 1.

All input data are fixed and only the provision costs are varied to investigate their influence on the optimal number of spare parts. Figure 2 presents the progression of the total costs at different provision costs in relation to the number of available spare parts.

The curve progression of the four different cases is similar. It is apparent that the total costs are changing significantly depending on the provision costs in contrast to the optimal number of spare parts. This results in a great scope for the provider of spare parts when setting the provision costs. Furthermore it can be seen that the total costs are determined by the provision costs, which is even more applicable when

Table 1 Input data to apply the model

Number of installed components			Provision costs for one spare part		Weighting of probability resulting from sensor data
10			Varied		50%
Component	Machine	Downtime costs of the machine	Probability of default (sensor data)	Probability of default (empirical values)	Effect on machine breakdown
1	1	20,000	0.02	0.01	0.8
2	1	20,000	0.06	0.05	0.8
3	2	30,000	0.20	0.07	0.8
4	2	30,000	0.08	0.70	0.7
5	3	15,000	0.09	0.11	0.5
6	3	15,000	0.80	0.13	0.3
7	4	35,000	0.11	0.05	1.0
8	4	35,000	0.12	0.17	0.2
9	5	25,000	0.13	0.11	0.6
10	5	25,000	0.14	0.21	0.9

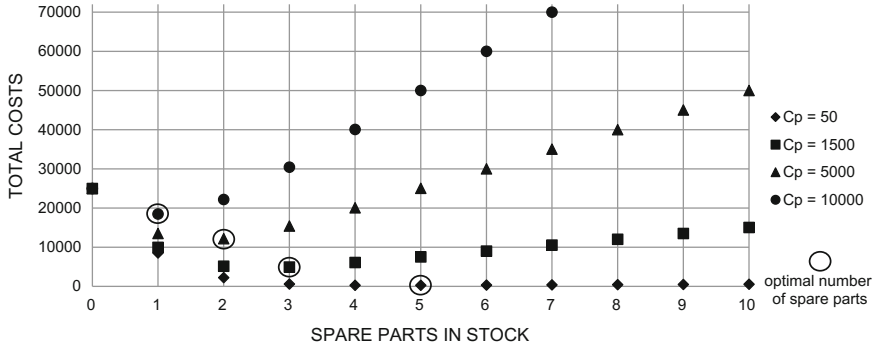


Fig. 2 Comparison of different provision costs in relation to the number of available spare parts

more spare parts are available. This leads to an almost linear increase. However, the impact of provision costs is depending on the structure of the probability of default and the influence of machine downtimes.

4 Conclusion

In this article, the challenge of determining the optimal number of spare parts was discussed. A new service concept was presented with the option to adapt the number of available spare parts in each period. A model was developed to optimize the number of spare parts through minimizing the total costs. To obtain a valid result both sensor data and empirical values were considered to determine the probability of default.

However, the number of branches is rising exponentially by an increasing number of considered components which has a strong effect on the computation possibilities. The calculation of the number of spare parts needed when a high number of components is installed is a challenge that has to be met. When using the optimization model it has to be taken into account that the calculations are conducted based on probabilities received through sensor data and empirical values. The reliability of this data determines decisively the quality of the calculations. Furthermore, several assumptions were made to simplify the model. For instance, it is assumed that the probabilities resulting from sensor data and empirical values are known as well as the downtime costs of the different machines. In future versions of the model expansions to reduce the quantity of assumptions should be made. The presented optimization model provides the basis for further development of determining the optimal number of available spare parts.

References

1. Aronis, K.-P., Magou, I., Dekker, R., Tagaras, G.: Inventory control of spare parts using a Bayesian approach: a case study. Eur. J. Oper. Res. **154**, 730–739 (2004)
2. Chang, P.-L., Chou, Y.-C., Huang, M.-G.: A (r, r, Q) inventory model for spare parts involving equipment criticality. Int. J. Prod. Econ. **97**, 66–74 (2005)
3. Elwany, A.H., Gebraeel, N.Z.: Sensor-driven prognostic models for equipment replacement and spare parts inventory. IIE Trans. **40**(7), 629–639 (2008)
4. Jin, T., Liao, H.: Spare parts inventory control considering stochastic growth of an installed base. Comput. Ind. Eng. **56**, 452–460 (2009)
5. Louit, D., Pascual, R., Banjevic, D., Jardine, A.K.S.: Condition-based spares ordering for critical components. Mech. Syst. Signal Process **25**(5), 1837–1848 (2011)
6. Yang, K., Niu, X.: Research on the spare parts inventory. In: 16th International Conference on Industrial Engineering and Engineering Management, pp. 1018–1021 (2009)

Scheduling on Uniform Nonsimultaneous Parallel Machines

Liliana Grigoriu and Donald K. Friesen

Abstract We consider the problem of scheduling on uniform processors which may not start processing at the same time with the purpose of minimizing the maximum completion time. We provide a variant of the MULTIFIT algorithm which generates schedules which end within 1.382 times the optimal maximum completion time for the general problem, and within $\sqrt{6}/2$ times the optimal maximum completion time for problem instances with two processors. Experimental results suggest that our algorithm is a viable option for addressing this problem in practice.

1 Introduction and Related Work

In practical scheduling situations, the machines may not become available for processing at the same time, and due to the progress in technology between the times at which different machines were produced, they may process jobs at diverse speeds. We consider the problem of nonpreemptively scheduling a given set of tasks on uniform processors with nonsimultaneous machine available times in order to minimize the maximum completion time.

This problem is strongly NP-hard since it is a generalization of the multiprocessor scheduling problem. For scheduling on parallel machines in order to minimize the maximum completion time, the algorithm MULTIFIT of Coffman, Garey and Johnson [1] is one of the most studied. For same-speed processors which do not start simultaneously, Lee [2] and Chang and Hwang [3] give worst-case analyses for scheduling on nonsimultaneous parallel machines in order to minimize the maximum completion time when using LPT and MULTIFIT, respectively. Recently, the exact

L. Grigoriu (✉)
Fakultät Für Wirtschaftswissenschaften, Wirtschaftsinformatik Und Wirtschaftsrecht, Universität Siegen, Kohlbettstr. 15, 57068 Siegen, Germany
e-mail: liliana.grigoriu@uni-siegen.de

D.K. Friesen
Department of Computer Science, Texas A&M University, College Station, TX 77840-3112, USA
e-mail: friesen@cs.tamu.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_62

bound for scheduling using MULTIFIT on nonsimultaneous same-speed machines, was established by Hwang and Lim [4] to be $24/19$ (about 1.2632).

For uniform processors that start simultaneously, worst-case approximation bounds of 1.4 and respectively 1.382 for a MULTIFIT variant were obtained in Friesen and Langston [5] and Chen [6]. For two uniform processors, Burkard and He [7] derive a worst-case bound of $\sqrt{6}/2$ (about 1.2247) if the MULTIFIT loop is repeated enough times. When MULTIFIT is combined with LPT as an incumbent algorithm, they show that the worst case bound decreases to $(\sqrt{2}+1)/2$ (about 1.2071).

Approximation for scheduling on uniform nonsimultaneous machines to minimize the maximum completion time was previously considered in He [8], where the maximum completion time of LPT schedules was shown to be within $5/3$ times the optimal schedule's maximum completion time, and that the bound is better when there are only two machines. A PTAS for the case where the number of machines is constant can be found in [9]. For the case where there is at most one period of unavailability (downtime) on each machine, that must not necessarily occur at the beginning of the schedule, Grigoriu and Friesen [10] give a MULTIFIT-like algorithm, LMULTIFIT, the schedules of which have maximum completion times that are at most 1.5 times the end of an optimal schedule or 1.5 times the latest end of a downtime. The bounds that apply to the MULTIFIT variants presented in [5, 6] for scheduling on simultaneous uniform machines also apply to LMULTIFIT when scheduling on nonsimultaneous uniform machines. A proof for this can be found in [11] (and a later version of it in [9]), and we give an outline thereof in this work, while emphasizing the main ideas. We also present experimental results.

2 Using LMULTIFIT for Scheduling on Uniform Nonsimultaneous Machines

A problem instance is given by a set of machines $M = \{M_1, \ldots, M_m\}$ and a set of independent jobs $J = \{1, \ldots, n\}$. Machine M_i has speed factor s_i and can start to process jobs at time $r_i \geq 0$. On the slowest machine, job j has processing time p_j. We call p_j the *length* of job j. The processing time of job j on a machine M_i is p_j/s_i. We thus assume that the slowest machine has a speed factor of 1.

We address this problem by using the algorithm LMULTIFIT from [10] in the simplified form it takes when it is used for the special case of scheduling on uniform nonsimultaneous machines. Given a desired accuracy ϵ, it works as follows [9, 11]:

LMULTIFIT (ϵ, ub, lb)

(1) Choose suitable values for upper bound ub and lower bound lb for the maximum completion time of the schedule, e.g., $ub = \max_{i\in\{1,\ldots,m\}}(\frac{\sum_{j=1}^{n} p_j}{s_i} + r_i)$, and $lb = 0$;

(2) Order the jobs in nonincreasing order of p_j;
(3) Set the deadline to $b = \frac{ub+lb}{2}$;
(4) Order the machines in nondecreasing order of $s_i(b - r_i)$;
(5) FFD: Assign jobs in nonincreasing order of p_j (as determined in step (2)) on the first processor on which they fit;
(6) If all jobs were assigned save the schedule and decrease the upper bound (ub = b);
(7) Else increase the lower bound (lb = b);
(8) If $ub - lb \geq \epsilon$ loop back to step (3);

The upper bound chosen in step (1) should be at least the maximum completion time of a schedule the user can construct and the lower bound should be at most the maximum completion time of an optimal schedule. LMULTIFIT uses the algorithm FFD (*first fit decreasing*) to assign tasks to processors (step (5)).

The MULTIFIT variants for scheduling on simultaneous uniform machines from [5–7] order the time intervals available for scheduling tasks in the same way as LMULTIFIT when applied to these problems. However, these variants have their initial upper and lower bounds for the duration of the schedule defined as a part of the algorithm, whereas LMULTIFIT allows the user to define these bounds. The proofs in [5–7] do not use the fact that the upper and lower bounds are defined as a part of their MULTIFIT variant when showing worst-case approximation bounds, which leads to the conclusion that these bounds also apply to LMULTIFIT. In this work we call *worst case approximation bound* of an algorithm A for a problem Q the highest ratio between the maximum completion time of a schedule produced by A for a problem instance I of Q and that of an optimal schedule of I.

We next summarize a proof that LMULTIFIT, when applied to nonsimultaneous uniform processors, obeys the worst-case approximation bounds from [6, 7] for scheduling using MULTIFIT on simultaneous uniform processors. We assume that the algorithm MULTIFIT for simultaneous uniform processors considered in this work orders time intervals available for scheduling in nondecreasing order of the speed factors of their processors, like the variants from [5–7].

For a problem instance I we denote with opt_I the end of an optimal schedule of I. A main part of the proof of the worst-case approximation bound of 1.382 for the general case where the number of processors is arbitrary, as it results from bound shown in [6], can be found in [11], where the following statement shown:

Theorem 1 (Approximation bound *[11]*) *Assuming that:*

(a) q is a worst-case approximation bound for MULTIFIT when scheduling on simultaneous uniform processors, and that
(b) for any Instance I of scheduling on uniform simultaneous parallel machines and for any MULTIFIT deadline $b \geq q * opt_I$ a feasible schedule is returned by the FFD algorithm, q is also a worst-case approximation bound for LMULTIFIT for scheduling on nonsimultaneous uniform processors.

It has been shown in [1] that reducing the deadline of MULTIFIT from a deadline where a feasible schedule exists to a smaller deadline does not necessarily result in another feasible schedule for problem instances with at least three processors. As a consequence, proofs about MULTIFIT worst-case approximation bounds are likely to contain proofs of property (b) from Theorem 1, as is the case in [6, 7]. Thus Theorem 1 implies that LMULTIFIT has a worst-case approximation bound of 1.382 when scheduling on uniform nonsimultaneous parallel machines.

In [9], a variant of Theorem 1 which applies to problems with at most m machines is also proved. Together with the results from [7], this implies that LMULTIFIT has a worst-case approximation bound of $\sqrt{6}/2$ when scheduling on at most 2 uniform nonsimultaneous parallel machines.

The statement of Theorem 1 can be proved by contradiction. For this, we assumed that there is a *counterexample*, that is, a problem instance I and a deadline $b \geq q * opt_I$ for which FFD within the LMULTIFIT loop does not produce a feasible schedule. We define a *minimal counterexample* to be a counterexample with a minimal number of processors. Obviously, if there is a counterexample, there also is a minimal counterexample. Let $I = (M, J)$ be a minimal counterexample, let $m = |M|$ and let $b \geq q * opt_I$ be a deadline for which FFD within the LMULTIFIT loop does not generate a feasible schedule. Using the concept of a minimal counterexample it can be shown that:

$$opt_I > \max_{p \in M}(r_p) \tag{1}$$

We call *length* of a time interval available for scheduling tasks its duration times the speed factor of the processor on which it is. The main idea of the proof of Theorem 1 is to use the minimal counterexample I and the deadline $b \geq q * opt_I$ where FFD within the LMULTIFIT loop fails to schedule all tasks, in order to create an instance I' of the problem of scheduling on simultaneous uniform machines, where the lengths of the time intervals available for scheduling are the same for I with deadline b as they are for I' with deadline b. As a consequence, MULTIFIT also fails to schedule all tasks for I' with deadline b. Equation (1) results in the fact that no additional spaces are created by this construction, as b must be greater than the end of an optimal schedule, and thus can not be smaller than the start of the processing time of any processor according to (1). It can be shown that $opt_{I'} \leq opt_I$. This results in a contradiction to q being a worst-case approximation factor for MULTIFIT for scheduling on uniform simultaneous machines (which implies that $b < q * opt_{I'}$), as that would result in: $b < q * opt_{I'} \leq q * opt_I \leq b$.

3 Experimental Results

In order to experimentally measure the performance of our algorithm, we use lower bounds that are derived from the properties of each instance. Given an instance, we know that all processing times must fit before the end b of the schedule, and thus

$$\sum_{i=1}^{m}(\max(0, b - r_i))s_i \geq \sum_{j\in J} p_j,$$

which, assuming the processors are ordered in nondecreasing order of their ready times, that is, $\forall i \in \{1, \ldots, m\} : r_i \leq r_{i+1}$ in this ordering, implies that:

$$\exists q \in \{1, \ldots, m\} : b \sum_{i=1}^{q} s_i - \sum_{i=1}^{q} r_i s_i \geq \sum_{j\in J} p_j$$

$$\Leftrightarrow \exists q \in \{1, \ldots, m\} : b \geq \frac{\sum_{j\in J} p_j + \sum_{i=1}^{q} r_i s_i}{\sum_{i=1}^{q} s_i}$$

$$\Leftrightarrow b \geq \min_{q=1}^{m}\left(\frac{\sum_{j\in J} p_j + \sum_{i=1}^{q} r_i s_i}{\sum_{i=1}^{q} s_i}\right)$$

We generated instances with 2, 3, 5, 10, 15 and 30 processors, with fractional speed factors allowed between 1 and 5, with the slowest processor having speed factor 1, with job lengths that can take integer values between 0 and a constant MaxJob. The machine available times can take integer values between 0 and MaxJob for one set of instances, and values between 0 and MaxJob (we used MaxJob = 200) times the average number of tasks per machine for another set of instances. We generated instances with an average of 2, 3, 5, 10, 15, 30 and 50 jobs per machine. All values are generated using a uniform probability distribution in Python. Thus, each time an instance is generated, two parameters are used, the number of machines and the average number of jobs per machine. For each problem set, for each parameter tuple, 100 instances are generated. In Tables 1 and 2 the average and the worst

Table 1 Approximation factors for LMULTIFIT, where the maximum ready time is the same as the maximum processing time

$\|M\|$	2 jobs/ machine	3 jobs/ machine	5 jobs/ machine	10 jobs/ machine	15 jobs/ machine	30 jobs/ machine	50 jobs/ machine
2	1.025896 1.13408	1.014911 1.0735	1.008147 1.04756	1.001934 1.01002	1.000732 1.00594	1.000245 1.00129	1.000096 1.00057
3	1.035263 1.14637	1.015074 1.06251	1.006792 1.02619	1.002015 1.00786	1.000896 1.00261	1.000239 1.00105	1.000096 1.00032
5	1.028481 1.10762	1.013325 1.03651	1.005365 1.01495	1.001463 1.00495	1.000719 1.00238	1.0002 1.00052	1.000089 1.00023
10	1.018373 1.05363	1.008773 1.04028	1.003419 1.00889	1.001113 1.00302	1.000511 1.00163	1.000168 1.00033	1.000088 1.00013
15	1.014423 1.03671	1.006645 1.01426	1.002785 1.00799	1.000789 1.00176	1.000412 1.00091	1.000154 1.00025	1.000087 1.00014
30	1.007669 1.01573	1.00391 1.00999	1.00161 1.00249	1.000538 1.00078	1.000305 1.00052	1.000147 1.00021	1.00009 1.00012

Table 2 Approximation factors for LMULTIFIT, where the maximum ready time is the maximum processing time times the average number of jobs per processor

$\|M\|$	2 jobs/ machine	3 jobs/ machine	5 jobs/ machine	10 jobs/ machine	15 jobs/ machine	30 jobs/ machine	50 jobs/ machine
2	1.021748 1.14418	1.010174 1.06859	1.002774 1.02171	1.000954 1.01033	1.000312 1.00171	1.00011 1.00073	1.000036 1.00015
3	1.019345 1.11243	1.006942 1.0283	1.002532 1.01688	1.00067 1.00265	1.000333 1.00247	1.000073 1.00041	1.000031 1.00014
5	1.01877 1.12979	1.006211 1.02614	1.002511 1.01046	1.000545 1.00198	1.00023 1.001	1.000067 1.00023	1.000027 1.00008
10	1.011845 1.03491	1.00434 1.01386	1.001345 1.00378	1.000327 1.00107	1.000165 1.00042	1.000051 1.00019	1.000025 1.00005
15	1.008729 1.03341	1.002911 1.01034	1.000955 1.00393	1.000268 1.00077	1.000124 1.00029	1.000046 1.00012	1.000027 1.00005
30	1.00468 1.01159	1.001866 1.0045	1.000674 1.00248	1.000193 1.00035	1.000101 1.00018	1.000046 1.00008	1.000028 1.00005

encountered approximation factors are listed. To calculate the approximation factor, we divide the end of the schedule found by LMULTIFIT by the lower bound of the instance as described above, that is, $\min_{q=1}^{m}(\frac{\sum_{j\in J} p_j+\sum_{i=1}^{q} r_i s_i}{\sum_{i=1}^{q} s_i})$.

We ran the same experiment with a maximum speed factor of 10, and then with a maximum speed factor of 10 and a maximum job length of 2000 and obtained similar results. The experiments suggest that LMULTIFIT performs very well in the average case, and that it is thus a viable option for addressing our problem in practice. The considered instances can be found at http://www.wiwi.uni-siegen.de/dekanat/kontakt/grigoriu/instancesnonsim/insts_nonsim.zip.

References

1. Coffman Jr., E.G., Garey, M.R., Johnson, D.S.: An application of Bin-Packing to multiprocessor scheduling. SIAM J. Comput. **7**(1), 1–17 (1978)
2. Lee, C.Y.: Parallel Machine Scheduling with nonsimultaneous machine available time. Discrete Appl. Math. **30**(1), 53–61 (1991)
3. Chang, S.Y., Hwang, H.: The worst-case analysis of the MULTIFIT algorithm for scheduling nonsimultaneous parallel machines. Discrete Appl. Math. **92**, 135–147 (1999)
4. Hwang, H.C., Lim, K.: Discrete Appl. Math. **167**, 172–187 (2014)
5. Friesen, D.K., Langston, M.A.: Bounds for multifit scheduling on uniform processors. SIAM J. Comput. **12**(1), 60–69 (1983)
6. Chen, B.: Tighter bound for MULTIFIT scheduling on uniform processors. Discrete Appl. Math. **31**(3), 227–260 (1991)
7. Burkard, R.E., He, Y.: A note on MULTIFIT scheduling for uniform machines. Computing **61**(3), 277–283 (1998)
8. He, Y.: Uniform machine scheduling with machine available constraints. Acta Math. Appl. Sin. (English Series), **16**(2), 122–129 (2000)

9. Grigoriu, L., Friesen, D.K.: Approximation for scheduling on uniform nonsimultaneous parallel machines. J. Sched. (2016). doi:10.1007/s10951-016-0501-1
10. Grigoriu, L., Friesen, D.K.: Scheduling on uniform processors with at most one downtime on each machine. Discrete Optim. **17**, 14–24 (2015)
11. Grigoriu, L.: Scheduling on parallel machines with variable availability patterns. Ph.D. thesis, Politehnica University Bucharest (2012)

Markov Models for System Throughput Analysis in Warehouse Design

Anja Heßler and Christoph Schwindt

Abstract In this paper we study the problem of computing the expected cycle time of a storage and retrieval system executing single-command cycles and being operated according to the closest open location rule. Assuming that the arrivals of the storage and the retrieval orders at the storage follow independent Poisson processes, we first consider the case of homogeneous inventory and develop closed-form expressions for the steady-state probabilities of a given storage location being selected for storage or retrieval. The approach is then generalized to storages with multiple stock keeping units. Comparing our results with estimation formulas from industry standards shows that the latter tend to significantly underestimate the maximum throughput of storage and retrieval systems.

1 Introduction

In warehouse design, appropriately dimensioning the storage and retrieval (S/R) system presupposes an accurate model of the system throughput under steady-state conditions. The expected maximum system throughput, calculated from the reciprocal expected cycle time, is largely influenced by the storage and retrieval strategy, which defines the way in which storage and retrieval orders are executed during warehouse operation. Given a set of orders to be processed, the strategy partitions the order set into operation cycles of the S/R system and allocates appropriate storage locations to each order. Disregarding the time savings achieved by optimally assigning storage locations to storage and retrieval orders may heavily bias the throughput analysis.

Based on continuous-time Markov chains we derive analytical results for the expected cycle time of S/R systems. We consider a rack storage under random storage location strategy serviced by rack feeders performing single-command cycles.

A. Heßler (✉) · C. Schwindt
Operations Management Group, Clausthal University of Technology,
Julius-Albert-Straße 2, 38678 Clausthal-Zellerfeld, Germany
e-mail: anja.hessler@tu-clausthal.de

C. Schwindt
e-mail: christoph.schwindt@tu-clausthal.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_63

We assume that storage and retrieval orders are released according to independent Poisson processes with arrival rates λ and μ, respectively, and are executed in the sequence of their arrivals. In the first setting (Sect. 2), we investigate the case of a homogeneous inventory of a single stock keeping unit (SKU), which serves us as a starting point to the analysis of the general case with multiple SKUs in Sect. 3. We further suppose that all orders refer to single loading units like pallets and that each storage location can hold exactly one loading unit of an arbitrary SKU. The storage and retrieval strategy considered in the following models is the closest open location rule often used in practice [5], which for each arriving order selects a storage location with minimum cycle time. In the case of homogeneous inventory, this storage and retrieval strategy maximizes the expected system throughput. For what follows, we assume that the N storage locations $n = 1, \ldots, N$ are numbered according to non-decreasing cycle times.

The remainder of this paper is organized as follows. In Sects. 2 and 3 we develop the Markov models for the cases of homogeneous inventory and multiple SKUs, respectively. In Sect. 4 we then compare the results of our models with alternative approaches to system throughput analysis from the literature and conclude the paper with a short summary and some remarks on future research avenues in Sect. 5.

2 The Case of Homogeneous Inventory

Let $\{Y(t) \mid t \geq 0\} = \left\{\left(Y_1(t), \ldots, Y_N(t)\right) \mid t \geq 0\right\}$ be the stochastic process with state space $E = \{0, 1\}^N$ modeling the evolution of the inventory distributed over the storage locations, where $Y_n(t)$ denotes the Bernoulli variable that equals 1 if storage location n is occupied at time t, and 0 otherwise. The possible state transitions directly follow from the closest open location rule. For each arriving order, the first feasible location in sequence $n = 1, \ldots, N$ of the storage locations is selected, i. e., for a storage order the first free and for a retrieval order the first occupied location is chosen. The respective transition rates correspond to the arrival rates λ and μ of storage and retrieval orders. The following proposition follows from the Poisson nature of the arrival processes, the finiteness of the state space, and the observation that each state $i \in E$ can be reached from the empty state $(0, \ldots, 0)$ by a sequence of state transitions and vice versa.

Proposition 1 *$\{Y(t) \mid t \geq 0\}$ is a homogeneous and irreducible continuous-time Markov chain.*

Consequently, the limiting distribution $\lim_{t\to\infty} (P(Y(t) = i))_{i\in E}$ exists and coincides with the unique stationary distribution $\pi = (\pi_i)_{i\in E}$ of the process, see, e.g., [6, p. 263]. We obtain the following global balance equations for π:

$$\pi_{(0,\ldots,0)} \cdot \lambda = \sum_{j \in E_1} \pi_j \cdot \mu, \qquad \pi_{(1,\ldots,1)} \cdot \mu = \sum_{j \in E_{N-1}} \pi_j \cdot \lambda \tag{1}$$

$$\pi_i \cdot (\lambda + \mu) = \sum_{n=1}^{n_o(i)-1} \pi_{i+e_n} \cdot \mu + \sum_{n=1}^{n_f(i)-1} \pi_{i-e_n} \cdot \lambda \quad (i \in E') \tag{2}$$

Let $E_m \subseteq E$ be the set of states with exactly m occupied locations. Equation (1) refer to the extremal states $(0, \ldots, 0)$ and $(1, \ldots, 1)$ corresponding to the empty and the full storage. State $(0, \ldots, 0)$ is left with rate λ by executing a storage order and reached with rate μ from every state $j \in E_1$ by executing a retrieval order. Symmetrically, state $(1, \ldots, 1)$ is left with rate μ by executing a retrieval order and reached with rate λ from every state $j \in E_{N-1}$ by executing a storage order. All other states $i \in E' = E \setminus \{(0, \ldots, 0), (1, \ldots, 1)\}$ considered in Eq. (2) can be left either by a storage or a retrieval order. With e_n denoting the nth unit vector and $n_o(i)$ and $n_f(i)$ being the first occupied and free storage location of state i, respectively, state i can be reached from every state $i + e_n$ with one more occupied location among the first $n_o(i) - 1$ storage locations by executing a retrieval order and from every state $i - e_n$ with one more free location among the first $n_f(i) - 1$ storage locations by executing a storage order.

We implemented the power method to solve this system of linear equations as recommended by Stewart [6, pp. 301ff.]. Due to the exponential increase of the state space's dimension in the number N of storage locations, we were only able to analyze small storages with $N \leq 19$. As we will show, however, we can calculate the expected cycle time based on truncated and aggregated versions of process $\{Y(t) \mid t \geq 0\}$, for which closed-form solutions of the stationary distributions can be specified. We are interested in the probabilities of a given storage location n being assigned to a storage or to a retrieval order. For fixed $n \in \{0, \ldots, N\}$, consider the stochastic process $\{Z^{(n)}(t) \mid t \geq 0\}$ with random variables $Z^{(n)}(t)$ counting the number of occupied locations among the first n storage locations. For this stochastic process, the following proposition holds.

Proposition 2 *For each $n \in \{1, \ldots, N\}$, process $\{Z^{(n)}(t) \mid t \geq 0\}$ corresponds to the throughput process of an $M/M/1/n$ queueing system with arrival rate λ and service rate μ.*

Using the formulas for the stationary distribution $\pi_k^{(n)}$ of an $M/M/1/n$ queueing system with utilization $\rho = \frac{\lambda}{\mu}$ and capacity n (see, e.g., [4, p. 79]), we obtain the probabilities $P_S(n)$ and $P_R(n)$ for storage location n of being chosen for a storage or a retrieval order in the following way. Storage location n is selected for a storage order precisely if storage locations 1 to $n - 1$ are occupied and the nth storage location is free. This holds true exactly if the first $n - 1$ but not the first n locations are occupied. As a consequence, $P_S(n) = P(\text{first } n - 1 \text{locations occupied}) - P(\text{first } n \text{ locations occupied})$. According to the PASTA property (Poisson arrivals see time averages, see [6, p. 394]), arriving orders see the stationary distribution, which provides

$$P_S(n) = \pi_{n-1}^{(n-1)} - \pi_n^{(n)} = \frac{(1-\rho)^2 \cdot \rho^{n-1}}{(1-\rho^n)\left(1-\rho^{n+1}\right)}$$

For the retrieval probability $P_R(n)$ of storage location n, we can apply similar arguments. Storage location n is selected for a retrieval order precisely if storage locations 1 to $n-1$ are free and the nth storage location is not.

$$P_R(n) = \pi_0^{(n-1)} - \pi_0^{(n)} = \rho \cdot P_S(n)$$

Note that probabilities $P_S(n)$ und $P_R(n)$ refer to *arriving* orders, which do not necessarily *enter* the system. If the storage is full or empty, storage and retrieval orders cannot be served, respectively, and are assumed to be lost. Consequently, it holds that $\sum_{n=1}^{N} P_S(n) = 1 - \pi_N^{(N)} < 1$ and $\sum_{n=1}^{N} P_R(n) = 1 - \pi_0^{(N)} < 1$.

3 The Case of Multiple SKUs

The aggregation approach presented in the previous section can be generalized to the case of multiple SKUs $\ell = 1, \ldots, L$ with arrival rates λ_ℓ and μ_ℓ of their storage and retrieval orders. For given $n \in \{0, \ldots, N\}$, we define the stochastic process $\{Z^{(n)}(t) \mid t \geq 0\} = \{(Z_1^{(n)}(t), \ldots, Z_L^{(n)}(t)) \mid t \geq 0\}$ with random variables $Z_\ell^{(n)}(t)$ counting the number of storage locations among the first n locations occupied by SKU ℓ at time t. By $E^{(n)} = \left\{ (j_1, \ldots, j_L) \mid \sum_{\ell=1}^{L} j_\ell \leq n \right\}$ we denote the state space of $\{Z^{(n)}(t) \mid t \geq 0\}$. From the Poisson property of the arrivals, the finiteness of the state space, and the reachability of all states $(j_1, \ldots, j_L) \in E^{(n)}$ we obtain

Proposition 3 *For each $n \in \{0, \ldots, N\}$, process $\{Z^{(n)}(t) \mid t \geq 0\}$ is a homogeneous and irreducible continuous-time Markov chain.*

We derive the stationary probabilities of $\{Z^{(n)}(t) \mid t \geq 0\}$ from the concept of detailed balance for Markov chains, see [6, p. 265], which splits the global balance equations into balance equations for any pair of states. For $\ell = 1, \ldots, L$ and $0 \leq \sum_{\ell=1}^{L} j_\ell \leq n-1$, we obtain the detailed balance equations

$$\lambda_\ell \cdot \pi^{(n)}_{(j_1,\ldots,j_\ell,\ldots,j_L)} = \mu_\ell \cdot \pi^{(n)}_{(j_1,\ldots,j_\ell+1,\ldots,j_L)} \tag{3}$$

The probabilities $\pi^{(n)}_{(j_1,\ldots,j_L)}$ solving the following system of product-form equations with $\rho_\ell = \frac{\lambda_\ell}{\mu_\ell}$ satisfy the detailed balance equations of Eq. (3).

$$\pi^{(n)}_{(j_1,\ldots,j_L)} = \pi^{(n)}_{(0,\ldots,0)} \cdot \prod_{\ell=1}^{L} \rho_\ell^{j_\ell} \quad ((j_1, \ldots, j_L) \in E^{(n)})$$

$$\sum_{k=0}^{n} \sum_{\substack{j_1,\ldots,j_L: \\ j_1+\cdots+j_L=k}} \pi^{(n)}_{(0,\ldots,0)} \cdot \prod_{\ell=1}^{L} \rho_\ell^{j_\ell} = 1$$

As detailed balance implies global balance, see [6, p. 249], $\pi^{(n)}$ is the unique stationary distribution of $\{Z^{(n)}(t) \mid t \geq 0\}$. Again, we can exploit an analogy to elementary concepts of queueing theory. For a closed Jackson network with L single-server stations and n circulating customers in the network, the global balance equations admit a similar product-form solution (see, e.g., [4, p. 196]). In difference to the closed queueing network, however, the number of stored units may be smaller than n. The probability $\pi^{(n)}_{(0,\ldots,0)} = \left[\sum_{k=0}^{n} \sum_{j_1,\ldots,j_L : j_1+\cdots+j_L=k} \prod_{\ell=1}^{L} \rho_\ell^{j_\ell}\right]^{-1}$ of an empty storage can be computed recursively in $O(L^2 n^2)$ time invoking the convolution algorithm of Buzen [2] for the normalizing constant of a closed Jackson network with single-server stations for each $k = 1, \ldots, n$. Once the stationary distribution $\pi^{(n)}$ is known, the storage and retrieval probabilities $P_S(n)$ and $P_R(n)$ arise from

$$P_S(n) = \sum_{\substack{j_1,\ldots,j_L: \\ j_1+\cdots+j_L=n-1}} \pi^{(n-1)}_{(j_1,\ldots,j_L)} - \sum_{\substack{j_1,\ldots,j_L: \\ j_1+\cdots+j_L=n}} \pi^{(n)}_{(j_1,\ldots,j_L)} \tag{4}$$

$$P_R(n) = \frac{1}{\sum_{\ell=1}^{L} \mu_\ell} \sum_{\ell=1}^{L} \mu_\ell P_R(n,\ell) \text{ with } P_R(n,\ell) = \sum_{\substack{j_1,\ldots,j_L: \\ j_\ell=0}} \left(\pi^{(n-1)}_{(j_1,\ldots,j_L)} - \pi^{(n)}_{(j_1,\ldots,j_L)}\right) \tag{5}$$

According to Eq. (5), the retrieval probability $P_R(n)$ is equal to the weighted mean of the retrieval probabilities $P_R(n,\ell)$ for SKUs ℓ. Again, using Buzen's convolution algorithm, the right-hand sides of Eqs. (4) and (5) can be calculated efficiently. In analogy to the single-SKU case, it can be shown that $P_R(n,\ell) = \rho_\ell \cdot P_S(n)$ and hence $P_R(n) = \rho \cdot P_S(n)$ with $\rho = \sum_{\ell=1}^{L} \lambda_\ell / \sum_{\ell=1}^{L} \mu_\ell$.

4 Numerical Example

The expected cycle time highly depends on the storage and retrieval strategy. Nevertheless, basic approaches to throughput analysis implicitly assume non-optimized, random allocations, leading to identical storage and identical retrieval probabilities $P_S(n)$ and $P_R(n)$ for all storage locations n, see, e.g., the handbook [1, pp. 659f.] or the industry standards [3, 7]. Hausman et al. [5] argue under restrictive assumptions that this uniform distribution can serve as a good approximation for $P_S(n)$ and $P_R(n)$ under the closest open location rule. In this section, we compare the expected cycle times arising from the uniform distribution assumption (UD) and our Markov model (MM(L), with L as the number of SKUs) for the closest open location rule.

In our example, we examine one aisle of a rack storage with 10 rows and 60 storage locations in each row. Assuming that the rack feeder can move simultaneously in horizontal and vertical direction, the travel times between the storage locations are determined using the Tchebychev metric. Furthermore, we suppose that $\lambda_\ell = \mu_\ell$ and hence $\rho_\ell = 1$ for all SKUs ℓ, which is generally satisfied by real-world storages operating under steady-state conditions. Table 1 displays the results for the different models.

Table 1 Comparison of expected cycle times

Model	MM(1)	MM(3)	MM(5)	MM(10)	MM(15)	UD
Expected cycle time	3.12	5.34	6.86	9.61	11.69	60.55

The results show that the model assuming uniform distribution heavily underestimates the maximum system throughput when the storage system is operated under the closest open location strategy. As a consequence, applying the industry standards may lead to a largely oversized system. We also notice that the expected cycle time significantly increases and hence the error of the UD model decreases when the number L of SKUs augments. In particular, it can be shown for $\rho_\ell = 1$ for all SKUs ℓ that as L tends to infinity, the limits of the storage probabilities $P_S(n)$ and $P_R(n)$ coincide and are identical for all $n = 1, \ldots, N$.

5 Conclusion and Future Work

In this paper we developed Markov models for the system throughput analysis of S/R systems executing single-command cycles under the closest open location strategy. We showed how the expected cycle time can be calculated efficiently based on aggregate models and exploiting analogies to finite-capacity service stations and closed Jackson networks. Comparing our results with traditional approaches assuming random selection of storage locations reveals that the latter may significantly underestimate the maximum system throughput. Our future research will be concerned with extending the models to dual-command cycles and considering batch arrivals of the storage and retrieval orders.

References

1. Arnold, D., Isermann, H., Kuhn, A., Tempelmeier, H., Furmans, K.: Handbuch Logistik, 3rd edn. Springer, Berlin (2008)
2. Buzen, J.P.: Computational algorithms for closed queueing networks with exponential servers. Commun. ACM **16**(9), 527–531 (1973)
3. Fédération Européenne de la Manutention FEM-Richtlinie 9.851: Leistungsnachweis für RBG-Spielzeiten. VDMA-Verlag, Frankfurt a. M (2003)
4. Gross, D., Shortle, J.F., Thompson, J.M., Harris, C.M.: Fundamentals of Queueing Theory. Wiley Series in Probability and Statistics, 4th edn. Wiley, Hoboken (2008)
5. Hausman, W.H., Schwarz, L.B., Graves, S.C.: Optimal storage assignment in automatic warehousing systems. Manage. Sci. **22**(6), 629–638 (1976)
6. Stewart, W.J.: Probability, Markov Chains, Queues, and Simulation: The Mathematical Basis of Performance Modeling. Princeton University Press, Princeton (2009)
7. Verein Deutscher Ingenieure: VDI-Richtlinie 3561: Testspiele zum Leistungsvergleich und zur Abnahme von Regalförderzeugen. VDI-Verlag, Düsseldorf (1973)

Lot Sizing and Scheduling for Companies with Tooling Machines

Florian Isenberg and Leena Suhl

Abstract The growing globalization and the rapid technical development intensify competition for nearly all manufacturing companies and increase the pressure to act. For companies using tooling machines to process metal, there is only little potential to improve the processing itself. A holistic view on the production system, however, provides an opportunity for cost savings and optimization. Therefore a two-level concept for lot sizing and scheduling is presented, transferring two different lot sizing and scheduling models from the literature into an integrated one. Each of the two models has a different time scope and an adjusted level of detail. The solution behavior and solution quality is analyzed for different test instances, and the advantages and disadvantages of such a two-level concept are pointed out.

1 Introduction

The increasing globalization and the rapid technology developments hold many different risks and opportunities for nearly all manufacturing companies. An intensified competition, new markets and additional competitors force these companies to act. This applies to companies with tooling machines as well. Especially for the small and medium-sized companies of this industrial sector, it is essential to be aware of these changes and to make full use of opportunities that already exist.

One possibility to increase the efficiency of the production is to improve the machine efficiency. However, in many cases the technical and economic limits of these machines are reached. The alternative is an efficiency improvement by optimizing the planning procedure.

F. Isenberg (✉) · L. Suhl
University of Paderborn, Warburger Str. 100, 33098 Paderborn, Germany
e-mail: isenberg@dsor.de

L. Suhl
e-mail: suhl@dsor.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_64

Tooling machines provide a solid basis for this kind of optimization. The flexibility is an advantage and can be used to produce many different products on one machine or one product on alternative machines, as long as the nc-program is present. This flexibility can also help to cope with unforeseen events, such as machine breakdowns or changes in the capacity or order situation. The goal of this paper is to solve an integrated lot sizing and scheduling problem for such companies, regarding long setup and processing times. Therefore, two different models from the literature are combined into an integrated one, which creates a natural decreased degree of detail within the planning horizon. This has the advantage of reducing the computational effort for later periods, especially if the productions in these periods are more likely to be changed before implementation.

The remainder of this paper is organized as follows. Section 2 gives a short overview of related literature. In Sect. 3 the mathematical model of the underlying problem is formulated. Numerical results for some test instances are discussed in Sect. 4 and the paper closes with a conclusion in Sect. 5.

2 Related Literature

The existing literature about multi-level lot sizing and scheduling with parallel machines is relatively scarce. Özdamar and Barbarosoğlu [7] model a production with serially-arranged manufacturing stages and parallel facilities. They use a CLSP kind of model to formulate the problem and extend it to include the different stages and loading decisions. The problem is solved by a combination of simulated annealing (SA), genetic algorithms (GA) and a Lagrangean relaxation scheme. In 2000 Belvaux and Wolsey [1] developed the "bc—prod modeling and optimization system". They present a generic model with a multi-level structure and production on multiple machines capable to produce one or more items per period. Test sets of different adapted models are solved by the system using, e.g., preprocessing and cutting plane techniques. Lead times are not considered. Kimms and Drexl [6] already introduced a proportional lot sizing and scheduling model (PLSP) with multi-level structure and parallel machines in 1998. The formulated model did not include setup times. For solution purposes a randomized regret-based sampling method is presented in the paper, but only evaluated for a model with multiple machines. The reviews of [2, 3] give a good overview about further literature on the subject.

The idea behind this work is not entirely new. Araujo et al. [5] already stated that it could be beneficial to use a simplified representation for later periods, as they are more likely to be not implemented. In contrast to other works this will be done by combining two different models here. A similar approach to use different models for a different time horizon can be found in [4].

3 Problem Statement

We combine a "multi-level continuous setup lot sizing problem" (MLCSLP) (detailed scope) with a "multi-level capacitated lot sizing problem with linked lot sizes" (MLCLSP-L) (rough scope). Both models include parallel resources.

Figure 1 shows the combined model and illustrates the different levels of detail. The model is formulated using the notation in Table 1.

Model:

$$Min \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}_j}\sum_{t\in\mathcal{T}}(s_{jm}\cdot st_{jm})\cdot x_{jmt} - \sum_{j\in\mathcal{J}}\sum_{m\in\mathcal{M}_j}\sum_{t\in\mathcal{T}_R}(s_{jm}\cdot st_{jm})\cdot z_{jmt} \tag{1}$$
$$+\sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} h_{jt}\cdot I_{jt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}} bc_{jt}\cdot r_{jt} + \sum_{m\in\mathcal{M}}\sum_{t\in\mathcal{T}} oc_m\cdot O_{mt} + \sum_{j\in\mathcal{J}}\sum_{t\in\mathcal{T}_R} ec_j\cdot e_{jt}$$

$s.t.:$

$$I_{jt} = I_{j(t-1)} + \sum_{m\in\mathcal{M}_j} q_{jmt} + e_{jt} - d_{jt} + r_{jt} - r_{j(t-1)} - \sum_{i\in\mathcal{S}_j}\sum_{m\in M_i} a_{ji}\cdot q_{imt}$$
$$\forall j\in\mathcal{J}, t\in\mathcal{T} \tag{2}$$

$$I_{jt} \geq \sum_{i\in\mathcal{S}_j}\sum_{m\in\mathcal{M}_i}\sum_{\tau=t+1}^{Min(t+v_j,T)} a_{ji}\cdot q_{im\tau} \qquad \forall j\in\mathcal{J}, t\in\mathcal{T}_{\mathcal{D}}\cup\{0\} \tag{3}$$

$$\sum_{j\in\mathcal{J}_m}(y_{jmt}+ys_{jmt}) + y_{0mt} = 1 \qquad \forall m\in\mathcal{M}, t\in\mathcal{T}_D \tag{4}$$

$$y_{jmt} - y_{jm(t-1)} \leq x_{jmt} \qquad \forall j\in\mathcal{J}, m\in\mathcal{M}_j, t\in\mathcal{T}_D \tag{5}$$

$$y_{jm(t-1)} + ys_{jmt} \leq 1 \qquad \forall j\in\mathcal{J}, m\in\mathcal{M}_j, t\in\mathcal{T}_D \tag{6}$$

$$ys_{jm(t-1)} + \sum_{i\in\mathcal{J}_m: i\neq j} y_{imt} \leq 1 \qquad \forall j\in\mathcal{J}, m\in\mathcal{M}_j, t\in\mathcal{T}_D \tag{7}$$

$$ys_{jm(t-1)} + \sum_{i\in\mathcal{J}_m: i\neq j} ys_{imt} \leq 1 \qquad \forall j\in\mathcal{J}, m\in\mathcal{M}_j, t\in\mathcal{T}_D \tag{8}$$

$$y_{0mt} \leq y_{0m(t-1)} \qquad \forall m\in\mathcal{M}, t\in\mathcal{T}_D \tag{9}$$

$$p_{jm}\cdot q_{jmt} + ST_{jmt} \leq C_{mt} + O_{mt} \qquad \forall j\in\mathcal{J}, m\in\mathcal{M}_j, t\in\mathcal{T}_D \tag{10}$$

$$\sum_{j\in\mathcal{J}_m}(p_{jm}\cdot q_{jmt}) \leq (C_{mt}+O_{mt}) \qquad \forall m\in\mathcal{M}, t\in\mathcal{T}_R \tag{11}$$

$$q_{jmt} \leq (C_{mt}+OC_{mt})\cdot y_{jmt} \qquad \forall j\in\mathcal{J}, m\in\mathcal{M}_j, t\in\mathcal{T}_D \tag{12}$$

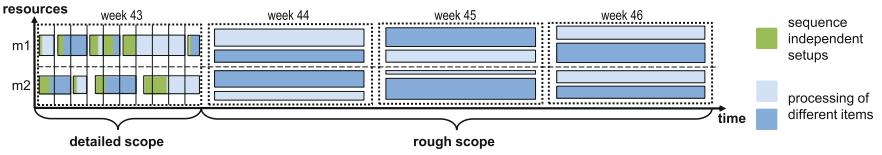


Fig. 1 Schematic representation of the two combined model

Table 1 Indices, index sets, parameters and decision variables of the model

Indices and index sets			
$\mathscr{T}$	Set of periods ($t \in \{1, \cdots, T\}$), with $\mathscr{T} = \mathscr{T}_D \cup \mathscr{T}_R$	$\mathscr{J}$	Set of items to produce
$\mathscr{T}_D \subseteq \mathscr{T}$	Set of detailed periods ($t \in \{1, \cdots, t_D\}$)	$\mathscr{J}_m$	Set of items produced by resource m
$\mathscr{T}_R \subseteq \mathscr{T}$	Set of rough periods ($t \in \{t_D + 1, \cdots, T\}$)	$\mathscr{S}_j$	Set of immediate successors of item j
$\mathscr{M}$	Set of resources	$\mathscr{M}_j$	Set of resources capable to produce item j

Parameters	
a_{ji}	Number of units of item j required to produce one unit of item i
d_{jt}	External demand of item j in period t
h_{jt}, bc_{jt}	Holding and backorder cost of one unit of item j in period t
s_{jm}	Setup cost of item j at resource m
oc_m	Overtime cost per unit at resource m
ec_j, v_j	Outsourcing cost per unit of item j and lead time of item j
p_{jm}, st_{jm}	Production time per unit and setup time of item j at resource m
C_{mt}, OC_{mt}	Capacity and overtime capacity of resource m in period t
Decision variables	
q_{jmt}	Production quantity of item j at resource m in period t
I_{jt}, r_{jt}	Inventory and backorder of item j at the end of period t, backorder is only allowed for end items
O_{mt}	Overtime at resource m in period t
e_{jt}	Outsourcing of item j in period t, outsourcing is only allowed for end items and rough periods
y_{jmt}	Setup state variable of item j at resource m at the end of period t, $y_{jmt} \in \{0, 1\}$
x_{jmt}	Finished setup process of item j at resource m in period t, $x_{jmt} \in \{0, 1\}$
ys_{jmt}	Setup process state variable of item j at resource m at the end of period t, $ys_{jmt} \in \{0, 1\}$
ks_{jmt}	Finished proportion of the setup process of item j at resource m at the end of period t, $ks_{jmt} \in [0, 1]$
ST_{jmt}	Performed setup time for item j at resource m in period t
int_{jmt}	Already finished units of item j at resource m at the end of period t, $int_{jmt} \in \{0, 1, 2, \dots\}$
sl_{jmt}	Finished proportion of the processed unit of item j at resource m at the end of period t, $sl_{jmt} \in [0, 1]$
z_{jmt}	Setup carry over of item j at resource m from period $t-1$ to t, $z_{jmt} \in \{0, 1\}$
v_{jmt}	Multiple period setup carry over of item j at resource m from period $t-1$ to $t+1$, $v_{jmt} \in \{0, 1\}$

$$q_{jmt} \leq (C_{mt} + OC_{mt}) \cdot x_{jmt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_R \tag{13}$$

$$KS_{jm(t-1)} + \frac{1}{st_{jm}} \cdot ST_{jmt} = x_{jmt} + KS_{jmt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_D \tag{14}$$

$$KS_{jmt} \leq 1 - \sum_{i \in \mathcal{J}_m} x_{imt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_D \tag{15}$$

$$KS_{jmt} \leq ys_{jmt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_D \tag{16}$$

$$y_{jmt} \leq 1 - \sum_{i \in \mathcal{J}_m : i \neq j} x_{imt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_D \tag{17}$$

$$ST_{jmt} \leq (C_{mt} + OC_{mt}) \cdot (x_{jmt} + ys_{jmt}) \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_D \tag{18}$$

$$int_{jmt} = int_{jm(t-1)} + sl_{jm(t-1)} + q_{jmt} - sl_{jmt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T} \tag{19}$$

$$int_{jm(t-1)} \leq int_{jmt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T} \tag{20}$$

$$sl_{jmt} \leq 1 - \sum_{i \in \mathcal{J}_m : i \neq j} x_{imt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_D \tag{21}$$

$$sl_{jmt} \leq 1 - \sum_{i \in \mathcal{J}_m : i \neq j} ys_{imt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_D \tag{22}$$

$$sl_{jmt} \leq z_{jm(t+1)} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_R \tag{23}$$

$$sl_{jm(t_D)} \leq z_{jm(t_D+1)} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j \tag{24}$$

$$\sum_{j \in \mathcal{J}_m} (z_{jmt}) \leq 1 \qquad \forall m \in \mathcal{M}, t \in \mathcal{T}_R \tag{25}$$

$$z_{jmt} \leq x_{jmt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_R \tag{26}$$

$$z_{jm(t_D+1)} \leq y_{jm(t_D)} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j \tag{27}$$

$$z_{jmt} \leq x_{jm(t-1)} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_R \setminus \{t_D + 1\} \tag{28}$$

$$z_{jmt} + z_{jm(t+1)} \leq 1 + v_{jmt} \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_R \setminus \{T\} \tag{29}$$

$$\sum_{i \in \mathcal{J}_m : i \neq j} x_{imt} \leq M \cdot (1 - v_{jmt}) \qquad \forall j \in \mathcal{J}, m \in \mathcal{M}_j, t \in \mathcal{T}_R \tag{30}$$

All decision variables non-negative. Overtime and outsourcing limited.

The objective is to minimize the holding, setup, overtime, outsourcing and backorder costs. The restrictions (2) and (3) ensure the inventory balance under consideration of the lead times. Backorder is only allowed for the end items and outsourcing only for the periods of the rough part of the model. The restrictions (4)–(9) deal with the setup state at the end of a period. Note that a dummy setup state is present and not all transitions are allowed. Inequalities (10)–(13) model the capacity restrictions and link production and setup. The restrictions (14) to (18) model setup processes over several adjacent periods. Setup times are only considered for the detailed part of the model. Due to the considered practical problem, the production lots should be integral, especially for items with a long processing time, guaranteed by the restrictions (19)–(24). The setup carry over of the MLCLSP-L is modeled by the inequalities (25)–(30). Furthermore, the connections between the two models are formulated.

An advantage of such a model is the natural decreasing level of detail and the separation between the two parts. It provides an opportunity for fast replanning of

Table 2 Test results

Name	J	M	T	$\#I$	(3, 10)		(3, 15)		(5, 10)		(5, 15)	
					$\overline{Gap_B}$ (%)	$\overline{Time_B}$ (%)	$\overline{Gap_B}$ (%)	$\overline{Time_B}$ (%)	$\overline{Gap_B}$ (%)	$\overline{Time_B}$ (%)	$\overline{Gap_B}$ (%)	$\overline{Time_B}$ (%)
C1B1a	8	2	16	2	66.00	0.84	66.84	0.35	0.00	13.86	0.00	4.29
C2B1	10	2	25	4	72.93	0.68	50.17	0.45	35.98	6.13	56.87	4.05
C2B2	10	3	25	2	48.82	0.64	8.16	25.49	17.98	23.57	17.58	25.05
C2B3	10	4	25	2	Infeasible		57.92	32.82	−0.95	25.03	−0.95	25.03
C3B1	10	2	52	4	71.75	38.43	77.12	5.04	19.34	45.55	67.44	21.14
C3B2	10	3	52	2	72.12	57.79	64.93	41.02	19.54	66.47	46.99	55.71
C3B3	10	4	52	2	15.78	53.64	Infeasible		50.74	87.53	45.90	51.83

the rough part, while the detailed one will be untouched. In addition, as little effort as possible is made for later periods. The savings can be used to increase the detail of the early periods. The disadvantage is an imprecise modeling of demands and the material flow in later periods, due to the bigger periods. Combined with the holding costs, this can lead to unmeant backorders or increased productions in later periods.

4 Numerical Tests

The combined model is evaluated with a small test set generated similar to the ones in [8]. A lead time of one period, backorder cost of 1000 per unit, a utilization of 0.7, a relative setup of 0.6 and a tbo of (1, 1) are set. The gozinto factors are rounded up. The instances, modeled completely with detailed periods, are used as a benchmark. Table 2 shows the results, performed on an Intel Core i7, with 2.2 Ghz and 16 GB main memory, using Gurobi 5.62. Only the two benchmark instances of C1B1a could be solved to optimality, the other ones were limited to four hours, with a remaining gap. Each line has $\#I$ instances with J items, M resources and T detailed periods. The combined model is evaluated within a rolling horizon, where, e.g., the parameter set $(3, 10)$ means, each iteration considers the next three detailed periods, while the other periods are united to rough periods, the size of up to 10 detailed periods. A time limit of one hour per iteration is set. The combined model is suitable to model the problem. For a few instances the last iteration was infeasible, due to unfortunate solutions in the previous iterations and the set lead times. The other results vary between poor solutions in almost four hours time and good or even improved solutions with substantial time saving. The increased costs can be explained by the missing information, leading to backorders in later periods. Therefore, the parameters have to be chosen carefully and evaluated further.

5 Conclusion and Outlook

In this paper a combination of two models with a different level of detail are presented and analyzed. The most important benefit is a reduced computational effort for later periods, especially when these will most likely be replanned. Apart from this there are some drawbacks. The modeling is more complicated and tends to cause additional costs in later periods due to lack of detail.

Further work can address the treatment of these drawbacks, especially the increased costs, the combination of different or more models or the comparison with one model including a decreasing level of detail. Further topics are the synchronization of the material flow for items with larger processing times or heuristic solution procedures making use of the special structure combining the two models.

Acknowledgements This work is supported by the German Federal Ministry of Education and Research as part of the Spitzencluster "its OWL".

References

1. Belvaux, G., Wolsey, L.A.: bc-prod: a specialized Branch-and-Cut system for Lot-Sizing problems. Manage. Sci. **46**(5), 724–738 (2000)
2. Buschkühl, L., Sahling, F., Helber, S., Tempelmeier, H.: Dynamic capacitated lot-sizing problems: a classification and review of solution approaches. OR Spect. **32**(2), 231–261 (2010)
3. Copil, K., Wörbelauer, M., Meyr, H., Tempelmeier, H.: Simultaneous lotsizing and scheduling problems: a classification and review of models. OR Spect. (2016). doi:10.1007/s00291-015-0429-4
4. Dangelmaier, W.: A concept for an accurate and closely coordinated production. In: Dangelmaier, W., Blecken, A., Delius, R., Klöpfer, S. (eds.) Advanced Manufacturing and Sustainable Logistics. Springer, Berlin (2010)
5. De Araujo, S.A., Arenales, M.N., Clark, A.R.: Joint rolling-horizon scheduling of materials processing and lot-sizing with sequence-dependent setups. J. Heuristics **13**(4), 337–358 (2007)
6. Kimms, A., Drexl, A.: Proportional lot sizing and scheduling: some extensions. Networks **32**(2), 85–101 (1998)
7. Özdamar, L., Barbarosoğlu, G.: Hybrid heuristics for the multi-stage capacitated lot sizing and loading problem. J. Oper. Res. Soc. **50**(8), 810–825 (1999)
8. Stadtler, H., Sahling, F.: A lot-sizing and scheduling model for multi-stage flow lines with zero lead times. Eur. J. Oper. Res. **225**(3), 404–419 (2013)

Flexible Production Scheduling with Volatile Energy Rates

Christoph Johannes, Matthias G. Wichmann and Thomas S. Spengler

Abstract The demand for electrical power in industrial production processes arises often in high energy costs for companies. In the future volatile energy rates, which are a consequence of the increasing power generation from renewable energies, can influence these energy costs. In order to reduce the energy costs with the help of volatile energy rates, latter have to be considered in the production scheduling. To date, only few planning approaches in the field of job-shop scheduling deal with volatile energy rates. A transfer into planning tasks of serial production as the economic lot scheduling problem is missing. This contribution introduces a planning approach for the energy-oriented lot sizing and scheduling problem.

1 Introduction

Energy is one of the most important inputs in manufacturing. In high energy-intensive industries such as paper manufacture the proportion of energy costs among the complete value added is about 18.7% and in middle energy-intensive industries such as metal products manufacture 9.1%, respectively. Even in low energy-intensive industries as automotive manufacture the share of energy costs among the value added is about 3.3%. The energy costs of all industries split on electrical power (50%), oil and natural gas (36%), coal (5%) and further energy sources (8%) [4]. As a consequence, companies demand for opportunities to reduce energy costs in order to compete in the world market.

C. Johannes (✉) · M.G. Wichmann · T.S. Spengler
Institute of Automotive Management and Industrial Production, Technische Universität Braunschweig, Mühlenpfordstr. 23, 38106 Braunschweig, Germany
e-mail: christoph.johannes@tu-bs.de

M.G. Wichmann
e-mail: ma.wichmann@tu-bs.de

T.S. Spengler
e-mail: t.spengler@tu-bs.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_65

Electrical power is a necessary and important input for most manufacturing systems. One main application of manufacturing systems is serial production. In serial production a periodic product mix of similar products is manufactured on one machine. Exemplary products are milling products such as different types of gears or formed products such as car body parts. To date, production planning in serial production focuses on the reduction of inventory and setup costs using approaches such as the economic lot scheduling problem (ELSP). Since in future time-dependent volatile energy rates are expected, the consideration of them is necessary in order to minimize production costs. By now, classical planning approaches do not account for decision-relevant energy costs.

In this paper we present a planning model for the flexible production scheduling with integrated lot sizing under consideration of volatile energy rates. In Sect. 2 the problem characteristics and the resulting impact on the constraints and on the objective function are described in detail. In Sect. 3 a modeling approach is discussed, which incorporates the aspects presented in Sect. 2. In Sect. 4 an illustrative example is given to show the reduction potential of production-related costs. The paper closes with a conclusion and an outlook.

2 Energy-Oriented Production Scheduling

In this section the problem characteristics of an energy-oriented lot sizing and scheduling problem (EOLSSP) are described. The task is to generate a feasible production schedule, meeting a given demand under consideration of machine capacities, with the aim of production-related cost minimization. In general, there are three characteristics: time-dependent volatile energy rates, fluctuating energy consumption of manufacturing machines and classical constraints of production scheduling.

The first characteristic concerns time-dependent volatile energy rates. Currently, the majority of electrical power is generated by classical power plants, which have a limited flexibility and are characterized by a constant supply of electrical energy. Therefore, electricity tariffs for companies are based on a constant energy and peak load rate. In the course of a sustainable power generation, most classical power plants will be replaced by renewable energies. This leads to two mayor changes. First, the energy supply of renewable energies fluctuates, as energy production from wind and photovoltaic is influenced by weather. Second, in order to meet current power demand, the fluctuating power supply is complemented by classical power plants. Based on the time-dependent demand and offer of electrical power, a power market with volatile energy rates arises. According to estimates, the new electricity tariffs will consist of a variable energy rate referring to the price of the power exchange market and a peak load rate limited on times with classical power generation [1]. As a result, companies are faced with volatile energy rates.

Second, as a consequence of time-dependent volatile energy rates, the manufacturing machines' time-dependent energy consumption has to be taken into account. The energy consumption is influenced by the machine state and load, whereby also in non-processing times a significant amount of energy is consumed [3, 6]. To date, lot sizing and scheduling approaches only determine the sequence of lots. Hereby, the determination of machine states' exact sequences with regard to idle times and interruptions in the production process is neglected (e.g. [2]). However, the sequence of machine states is necessary to determine the time-dependent energy consumption. In general, four types of machine states are distinguished. In the first machine state type 'shutdown' the manufacturing machine consumes few energy, but also manufactures no product and loses its setup. In the second machine state type 'idle' the manufacturing machine performs also no production process but consumes more energy than in the machine state 'shutdown', based on the operating of auxiliary units. The setup is not lost. In the third machine state type 'setup' the machine consumes energy for setting up. This machine state is needed as transition between different product setups or as starting-up from 'shutdown'. In the fourth machine state type 'manufacturing' the machine performs a production process, whereby the most energy is consumed. During the machine state 'manufacturing' the energy consumption depends on the machine load that is linked to the product being manufactured. Therefore, for each product a separate 'manufacturing' machine state is considered.

Third, conventional constraints concerning lot sizing and scheduling need to be respected. These constraints are the fulfillment of demand without backlogging, the balance of inventory and the consideration of setup and machine states. The majority of these constraints are considered in existing models (e.g. [5]).

The objective of planning is the minimization of relevant production-related costs. The production-related costs comprise inventory, setup and energy costs. While inventory and setup costs are regarded in classical planning approaches (e.g. [5]), energy costs have not been considered in this planning task before. The energy costs are based on time-dependent energy consumption and time-dependent energy rates. The time-dependent energy consumption is given by the sequence of machine states. The time-dependent energy rates are given by extern.

3 Model

In this section the specifics of the mathematical formulation for the EOLSSP are discussed. The objective function is presented and the constraints are described.

The introduced planning problem is a lot sizing and scheduling problem, i.e. transforming customer demand into production lots with the aim to minimize the production-related costs. We consider the case of a single machine. On this machine $p = 1, \ldots, P$ products have to be produced. The set of machine states S comprises four different types of machine states. These are the non-producing states s^{idle}, $s^{shutdown}$ and s^{setup}. Besides, for the manufacturing of each product p there is a separate machine state $s_p^{man.}$. The planning horizon comprises $t = 1, \ldots, T$ periods. Three

classes of binary and one class of continuous variables are used to formulate the model. The binary decision variable $x_{s,t}$ is set to one if the machine starts with the machine state in period t. The binary variable $y_{s,t}$ is set to one if the machine state s is active in period t. The third binary variable $z_{p,t}$ is set to one if the machine can manufacture product p in period t. The continuous variables $I_{p,t}$ describe the inventory of product p at the end of period t.

The objective function (1) minimizes the relevant production-related costs C_{total}, consisting of setup, inventory and energy costs. The setup costs arise from the amount of setups evaluated with the setup cost factor c^{setup}. The inventory costs arise from products on stock evaluated with the inventory cost factor $c_p^{inventory}$ in dependence on the product p. The energy costs for one period t arise from the active machine states $y_{s,t}$ evaluated with the energy consumption p_s in dependence on the machine state s and the volatile energy rate c_t^{energy}.

$$\min Z = C_{total} = \sum_{t=1}^{T} \left(y_{s^{setup},t} \cdot c^{setup} + \sum_{p=1}^{P} I_{p,t} \cdot c_p^{inventory} + \sum_{s \in S} y_{s,t} \cdot p_s \cdot c_t^{energy} \right) \tag{1}$$

Overall, five categories of constraints are considered in this modeling approach. The first three categories result from the consideration of the exact sequence of machine states and the production of various products, while the last two categories are classical restrictions in the course of production planning. The first category of constraints defines $z_{p,t}$. It is needed to restrict the theoretical possible products p to be manufactured in period t referring to the prior machine setup and state. So, if the machine manufactures a product in the prior period the machine will be able to only manufacture this product until the machine's setup is renewed. If the machine is transferred into the machine state 'shutdown', the setup will be lost. In the machine state 'idle' the machine keeps the setup of $z_{p,t-1}$ in period t. The second category of constraints defines $x_{s,t}$. While the machine states s^{idle}, $s^{shutdown}$ and s^{setup} are permitted at any time, machine state $s_p^{man.}$ in period t will be only permitted, if the product p according $z_{p,t}$ can be manufactured in period t. Third, the binary variable $y_{s,t}$ is determined by a new active machine state $x_{s,t}$ and the machine state of the prior period $y_{s,t-1}$. Hereby, it has to be ensured that exactly one machine state s is active in each period t. The fourth category of constraints defines the inventory stock $I_{p,t}$. It is determined by the inventory stock of the prior period $I_{p,t-1}$, the demand $d_{p,t}$ and in period t manufactured products given by $y_{s,t}$ and their output quantity of product p in dependence of the machine state s. The last category comprises the binary and nonnegativity constraints. They determine the range of decision variables. Also an initial machine state is given.

All mentioned constraints can be formulated as linear constraints. Therefore, the resulting model can be categorized as a MILP (mixed-integer linear problem). However, the model is hard to solve as several binary variables exist.

4 Illustrative Example

In this section an illustrative example to show the effects of considering energy costs is given. To achieve numerical results, the planning model was implemented in CPLEX 12.6.2 and solved on a 2.5 GHz CPU with 8 GB RAM.

In the example, a production plan for $T = 28$ periods regarding $P = 3$ products is determined. Thus, the set of machine states comprises 6 states, in which 3 are non-producing and 3 are producing states. The further relevant parameters are randomly generated based on real world values. The time-dependent volatile energy rates c_t^{energy}, measured in monetary units (MU) per kWh, are given in Table 1. The product-related characteristics as inventory costs $c_p^{inventory}$ per period and product p, measured in MU, the demand for products in various periods and the output quantity of products in dependence of the machine states s are given in Table 2. The manufacturing machine's energy consumption in dependence of their state s per period, measured in kWh, is given in Table 3. The setup costs c^{setup} are 150 MU per setup.

To obtain comparable results, the numerical example is solved in two different ways. First, following a classical ELSP approach, the objective function is reduced to the sum of setup and inventory costs. Second, following our EOLSSP approach, the objective function is kept as given in Formula (1). The obtained results are given in Fig. 1a for the sequence of machine states of the ELSP approach and in Fig. 1b for the sequence of machine states of the EOLSSP approach. The two solutions differ greatly from each other. There are three major findings. First, the total costs of the ELSP schedule are 2248 MU, while the one of the EOLSSP are only 2185 MU. Thus, in total, 2.9% of costs may be saved. Second, the number of setups decreases from 4 (ELSP) to 3 (EOLSSP). This is due to the fact that without the consideration of

Table 1 Time-dependent energy rates

Period	1	2	3	4	5	6	7	8	9	10	11	12	13	14
Energy rate	4.09	4.04	4.00	3.96	3.91	4.45	4.98	5.51	6.05	6.80	7.56	8.31	9.07	9.21
Period	15	16	17	18	19	20	21	22	23	24	25	26	27	28
Energy rate	9.35	9.50	9.64	9.26	8.88	8.50	8.12	7.84	7.56	7.28	7.00	6.84	6.68	6.52

Table 2 Product-related characteristics

Product	Inventory costs	Demand in period			Output quantity in machine state		
		18	26	28	$s_1^{man.}$	$s_2^{man.}$	$s_3^{man.}$
1	0.3	200	200	200	100		
2	0.5	160	160	160		80	
3	0.6	120	120	120			60

Table 3 Energy consumption in dependence of machine states

	Machine states					
	$s^{shutdown}$	s^{idle}	s^{setup}	$s_1^{man.}$	$s_2^{man.}$	$s_3^{man.}$
Energy consumption	1	2	4	6	8	10

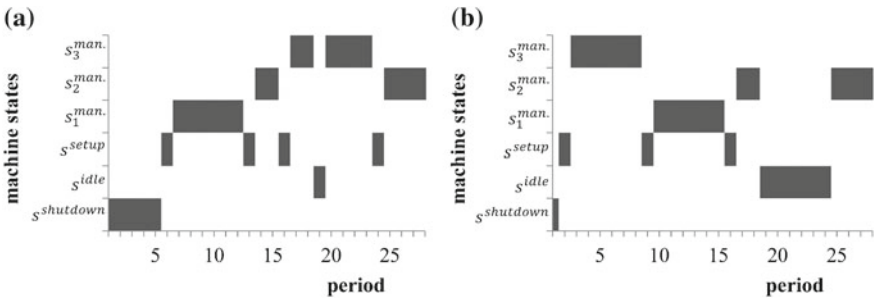


Fig. 1 Optimal solution schedule over 28 periods, **a** ELSP **b** EOLSSP

energy rates, it is useful to setup the machine for the production of product 2 twice. Third, inventory costs in the EOLSSP increase from 380 MU to 619 MU. Thus, a larger amount of products is stored for more periods, since production starts in early periods with low energy rates. Nevertheless, due to savings of energy costs, a positive impact is achieved. As a result, while being economically reasonable, the consideration of energy costs has a high impact on planning and leads to highly different production plans than classical approaches.

5 Conclusion and Outlook

In this paper we present the EOLSSP considering a machine's energy consumption and time-dependent volatile energy rates. The characteristics of the planning problem are discussed, transformed into an adequate planning approach and illustrated. As a result, the contribution of the paper is twofold. First, energy costs are considered within the EOLSSP. Second, we show a cost saving potential of 2.9%.

Future research should address three directions. First, there is an enormous need of relevant data regarding energy consumption in different machine states. Here, methods need to be applied in order to obtain the relevant data. Second, the model formulation is hard to solve due to the binary variables. Here, either a different formulation or a suitable solution procedure has to be developed. Third, the approach should be applied to larger case studies in order to verify and outline the potential of energy consideration in short term production planning.

References

1. Bundesministerium für Wirtschaft und Energie (ed.): Ein Strommarkt für die Energiewende: Ergebnispaper des Bundesministeriums für Wirtschaft und Energie. Berlin (2015)
2. Fleischmann, B., Meyr, H.: The general lotsizing and scheduling problem. OR Spektrum **19**(1), 11–21 (1997)
3. Liu, Y., Dong, H., Lohse, N., Petrovic, S., Gindy, N.: An investigation into minimising total energy consumption and total weighted tardiness in job shops. J. Clean. Prod. **65**, 87–96 (2014)
4. Matthes, F.C., Greiner, B., Neuhoff, K., Petrick, S., Ritter, N., Cook, V.: EKI–Der Energiekostenindex für die deutsche Industrie (2016)
5. Rogers, J.D.: A computational approach to the economic lot scheduling problem. Manag. Sci. **4**(3), 264–291 (1958)
6. Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., Ortega-Mier, M.: Optimizing the production scheduling of a single machine to minimize total energy consumption costs. J. Clean. Prod. **67**, 197–207 (2014)

Part XVI
Project Management and Scheduling

Audit Scheduling in Banking Sector

Ethem Çanakoğlu, İbrahim Muter and Onur Adanur

Abstract In this paper, we handle an audit scheduling problem in which the task requirements and the auditor experiences are quantified and forge a set of restrictions in team formation. We propose a mathematical model for this problem, propose two heuristics for its solution, and evaluate their performance through computational experiments.

1 Introduction

Internal auditing is a key function in financial organizations in order to evaluate and improve the effectiveness of risk management, control, and governance processes. It involves examination and evaluation of the activities carried out at the headquarters as well as local branches. The tasks of the branch audit program include the compliance check with organization policies and regulatory requirements. Depending on the size of a financial organization, the number of branches may range from a few to thousands. Therefore, efficient scheduling of the auditor workforce to the branches is an important problem for the internal auditing department because of the vital role of workforce in the performance of auditing operations, and its share in the total operational cost.

Audit scheduling is one of the prominent problems classified under staff scheduling (See [4] for extensive review of staff scheduling problems). This problem deals with the assignment of a set of personnel qualified as a team to a set of branches

E. Çanakoğlu (✉) · İ. Muter
Bahçeşehir University, Çırağan Cad. Osmanpaşa Mektebi Sok. No: 4-6,
34349 Besiktaş/Istanbul, Turkey
e-mail: ethem.canakoglu@eng.bau.edu.tr

İ. Muter
e-mail: ibrahim.muter@eng.bau.edu.tr

O. Adanur
Department of Industrial Engineering, Boğaziçi University,
34342 Bebek/Istanbul, Turkey
e-mail: onur.adanur@boun.edu.tr

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_66

to be audited under operational constraints. Since these problems are perplexed by the real-life operational constraints, they are generally solved by heuristic methods [2, 3]. In this paper, we handle an audit scheduling problem arising at a Turkish bank. The novel structure of this problem is that the auditor experience levels and the experience requirements of the branches are quantified and the total experience of each team is to satisfy the requirements of the assigned branches. The objective of this problem is to complete all the audit tasks as early as possible by using the available auditors. An important imposition in handling this problem is that members of a team work together until the end of the auditing horizon. Although this restriction leads to longer audit horizon, it not only facilitates the solution of the problem but also the performance evaluation of the team. We propose a mathematical model for this problem in Sect. 2, which has two decomposable aspects: the team formation and the branch scheduling. The former involves technical constraints associated with the size and total experience of a team, and the latter evokes an extension of the parallel machine scheduling problem with makespan minimization where the machines correspond to the teams. We propose two heuristics for the solution of the mathematical model that are explained in Sect. 3 and evaluate their performance through computational experiments presented in Sect. 4.

2 Model Description

The internal auditing department has a set of auditors indexed by $i \in I$. Each auditor $i \in I$ has experience level denoted by c_i ranging between 1 and 5 where 1 corresponding to the trainee-level auditors and 5 being the top auditors. The set of branches that need to be audited is denoted by B. Each branch $b \in B$ is associated with two parameters, the total experience requirement (r_b) and the duration of the audit (d_b). The disjoint groups of auditors satisfying a set of constraints form the audit teams indexed by $j \in J$, which are then assigned to the branches. The auditors with experience level 5 and some of those with level 4 constitute the senior auditor set $I_m \subset I$. Each team needs to have one senior auditor so that the maximum number of teams that can be formed are equal to the number of senior auditors.

As alluded to previously, the teams, once formed, stay intact throughout the planning horizon. Hence, the total experience of the auditors assigned to a team must be larger than or equal to the minimum experience requirement of the branches audited by it. Also, lower (L) and upper (H) bounds limit the number of auditors in a team.

For the integer programming formulation of this problem, we define binary decision variable y_j associated with $j \in J$ which indicates whether team j is formed or not, binary auditor assignment variable x_{ij} that takes value 1 only if auditor i is assigned to team j, and binary branch assignment variable z_{jb} which is equal to 1 only if group j is assigned to branch b. Since each team must be assigned a senior auditor, we set $|J| = |I_m|$ in the model, possibly causing some of the teams to be empty. The mathematical model is given as follows:

$$
\begin{aligned}
\mathscr{P}: \quad \text{Minimize} \quad & C_{max} \\
\text{Subject to} \quad & \sum_{j \in J} x_{ij} \leq 1, && \forall i \in I, \quad (1) \\
& \sum_{i \in I_m} x_{ij} \geq y_j, && \forall j \in J, \quad (2) \\
& L y_j \leq \sum_{i \in I} x_{ij} \leq H y_j, && \forall j \in J, \quad (3) \\
& \sum_{j \in J} z_{jb} = 1, && \forall b \in B, \quad (4) \\
& \sum_{i \in I} c_i x_{ij} \geq r_b z_{jb}, && \forall j \in J, b \in B \quad (5) \\
& \sum_{b \in B} d_b z_{jb} \leq C_{max}, && \forall j \in J, \quad (6) \\
& \sum_{b \in B} z_{jb} \leq M y_j, && \forall j \in J, \quad (7) \\
& x_{ij}, y_j, z_{jb} \in \{0, 1\} && \forall i \in I, \forall j \in J, \forall b \in B. \quad (8)
\end{aligned}
$$

Constraints (1) ensure that each auditor is assigned to at most one team. Constraint set (2) imposes all groups to include at least one senior auditor. If a team is formed, constraints (3) limit its size between L and H. While constraint set (4) forces each branch to be assigned to a group, constraints (5) ensure that all branches are assigned to teams with sufficient total experience level. Constraint set (6) determines the maximum completion time among the team schedules, referred as C_{max}, which is minimized in the objective function. The assignment of branches to an unformed team is prevented through constraint set (7), and finally, constraint set (8) imposes binary restrictions on the decision variables.

This problem has a decomposable structure that yields two problems, namely a team formation (1)–(3) and an extension of the parallel machine scheduling problem (4)–(6). For a given set of teams, the resulting problem is an extension of the parallel machine scheduling problem in which each job (branch) $b \in B$ is only allowed to be processed on subset of the parallel machines (teams). On the other hand, for a given set of schedules, the remaining problem boils down to finding feasible teams. However, in both cases, finding a feasible solution is not guaranteed. Next, we propose two heuristic methods to obtain a good upper-bound for this problem by exploiting the decomposable structure of $\mathscr{P}$. We form teams and schedules using a sequential approach in which the major difficulty is determining a limit on the total duration of branches assigned to a team. To that end, we determine an approximation of the makespan by applying the longest processing time (LPT) first rule on $|I_m|$ fictitious teams without considering constraints associated with team formation. Given this $\hat{C}_{max}$ value, (6) becomes a knapsack constraint. Next, we explain the proposed heuristics which differ in the way the branches are combined together to form a team schedule.

3 Heuristic Methods

The first heuristic, which is given in Algorithm 1 and is referred to as the knapsack-based heuristic, tries to form a team that is capable of auditing the unassigned branch with the largest requirement, and then, solves a knapsack problem to select a set of unassigned branches without considering the experience compatibility issues. To accelerate the operations, we initialize the unassigned branch set B in a non-decreasing order of the branch requirements r_b. A feasible team that has total experience larger than or equal to the maximum requirement of the unassigned branches, given as $\beta = r_{b_1}$, is sought by solving the assignment problem $\mathscr{P}_A$. We strive to form a team whose total experience satisfies the requirement in the minimal way by minimizing the surplus of constraint (13). If $\mathscr{P}_A$ is infeasible, (14) is relaxed by allowing more than one senior auditors for each team, which is more likely to give a feasible solution. However, a team may still not be formed, in which case the branch with the largest requirement b_1 is added to the previously generated schedule with the smallest duration due to the makespan objective. If a feasible team is formed, we solve the knapsack problem, tagged $\mathscr{P}_K$, to form a schedule consisting of uncovered branches including b_1. The objective of this problem is to maximize the total requirement of the selected branches, which is emphasized by the squares of r_b, $b \in B$. Such an objective prompts "hard" branches to be covered by teams at the early stages when β is generally large. Finally, if there still exist unassigned branches but no senior auditor remains, then these branches are allocated to the existing schedules using the LPT rule. Observe that at the end of the algorithm, the actual C_{max} value may be larger than $\hat{C}_{max}$.

$$\mathscr{P}_K : \text{maximize} \sum_{b \in B} r_b^2 z_b, \tag{9}$$

$$\text{s.t.} \sum_{b \in B} d_b z_b \leq \hat{C}_{max} - d_{b_1}, \tag{10}$$

$$z_b \in \{0,1\}, \ \forall b \in B. \tag{11}$$

$$\mathscr{P}_A : \text{minimize } s, \tag{12}$$

$$\text{s.t.} \sum_{i \in I} c_i x_i - s = \beta, \tag{13}$$

$$\sum_{i \in I_m} x_i = 1, \tag{14}$$

$$L \leq \sum_{i \in I} x_i \leq H, \tag{15}$$

$$x_i \in \{0,1\}, \ \forall i \in I. \tag{16}$$

The second algorithm, given in Algorithm 2, is reminiscent of the savings algorithm that was proposed in [1] for solving the vehicle routing problem. The algorithm makes use of a saving parameter defined for each pair of customers which correspond to the actual saving in distance if two customers are visited consecutively by the same vehicle. Similarly, we calculate the saving value for a pair of branches (i,j) as $S_{ij} = |r_i - r_j| - \max\{r_b : b \in \{i,j\}\} / \max\{r_b : b \in B\}$. The branch pairs are sorted in non-decreasing order of savings values since it is desirable that the branches with similar requirements reside in the same schedule. Moreover, the term subtracted ensures that the difficult branch pairs take smaller values so that they are prioritized in the schedule generation. Instead of solving a knapsack problem as in the first algorithm, we follow the list of branch pairs according to the savings values and concatenate an unassigned branch, say b', to the current schedule S if its pair exists in this schedule and the total duration of the schedule does not exceed $\hat{C}_{max}$. If the latter is not satisfied, we remove b' from the list of unassigned branches that can be added to S, which is denoted by S'. When no more branch can be added to S, we solve $\mathscr{P}_A$ to form a feasible team that can audit S. Unlike in the knapsack-based algorithm, if no feasible solution to $\mathscr{P}_A$ can be found even after relaxing (14), there is a possibility that the branch in S with the largest requirement may not be assigned to one of the previously formed teams. This may lead to an infeasible solution, though not encountered in the computational experiments.

Algorithm 1 Knapsack based

Apply LPT to calculate $\hat{C}_{max}$, sort B in non-decreasing order of r_b
while $B \neq \emptyset$ & idle seniors exist **do**
  solve $\mathscr{P}_A$ with $\beta = r_{b_1}$.
  if $\mathscr{P}_A$ is infeasible **then**
    Relax (14) and resolve $\mathscr{P}_A$.
  end if
  if $\mathscr{P}_A$ is feasible **then**
    solve $\mathscr{P}_K$.
    $B \leftarrow B \backslash N_1$, $N_1 = \{b : z_b = 1\}$.
    $I \leftarrow I \backslash N_2$, $N_2 = \{i : x_i = 1\}$.
  else
    Add b_1 to the schedule with the smallest duration.
  end if
end while
if $B \neq \emptyset$ & no idle seniors exist **then**
  Apply LPT rule.
end if

Algorithm 2 Savings based

Apply LPT to calculate $\hat{C}_{max}$, $S = \emptyset$
Sort $s_{bb'}$ for $b, b' \in B$ in non-decreasing order.
while $B \neq \emptyset$ && idle seniors exist **do**
  $(bb') = \arg\min(s)$, $S \leftarrow \{b, b'\}$, $d_S = d_b + d_{b'}$, $S' \leftarrow B \backslash \{b_b, b_{b'}\}$
  while $d_S \leq \hat{C}_{max} - \min(d_b : b \in B)$ **do**
    $b' = \arg\min(s_{bb'} : b \in S, b' \in S')$, $S' \leftarrow S' \backslash \{b'\}$
    if $d_s + d_{b'} \leq \hat{C}_{max}$ **then**
      $S \leftarrow S \cup b'$, $d_S \leftarrow d_S + d_{b'}$
    end if
  end while
  Solve $\mathscr{P}_A$ with $\beta = \max(r_b, b \in S)$.
  if $\mathscr{P}_A$ is infeasible **then**
    Relax (14) and resolve $\mathscr{P}_A$.
  end if
  if $\mathscr{P}_A$ is feasible **then**
    $I \leftarrow I \backslash N_2$, $N_2 = \{i : x_i = 1\}$, $B \leftarrow B \backslash S$
  else
    $b \leftarrow \arg\max(r_b, b \in S)$. Assign b. $B \leftarrow B \backslash \{b\}$
  end if
end while
if $B \neq \emptyset$ & no idle seniors exist **then**
  Apply LPT rule.
end if

4 Numerical Experiment

In this section, we test our proposed heuristics on ten randomly generated instances (set 1–10) for different problem sizes. The experiments were conducted with a 3.6 GHz Intel Xeon E5-1620 processor and 16 GB of RAM. The algorithms were implemented in C++, and the MIP solver of CPLEX 12.6 is used for exact solution of the models. We used a small instance set with 15 auditors and 50 branches in order to evaluate the performance of the heuristics against the optimal solution whereas for large instances with 200 auditors and 1300 branches, we compared the performance of the heuristics with lower and upper bounds found by CPLEX in 1 h time limit. For both problem sizes, we have varied the Pearson correlation coefficient of duration and requirement of branches between 0 (set 1) to 0.9 (set 10). We used the Gaussian copula model [5] to ensure correlation between the duration and requirement of branches. The auditor capabilities (between 1–5) are not changed through instances. The duration of the branches are generated from a discrete triangular (3,7.5,16) distribution and the requirement of the branches follow discrete triangular (6,12.5,17) distribution (Table 1).

The savings based method results in smaller C_{max} compared to knapsack based algorithm in all instances. The average gap between the optimal solution and savings based method is 1%, and that between optimal and knapsack based method is 9%. Knapsack based algorithm forms fewer teams with greater experience level yielding

Table 1 1300 branch 200 auditor results

		1	2	3	4	5	6	7	8	9	10
CPLEX (1 h)	Upper bound	261	258	259	266	267	261	260	259	250	258
	Lower bound	217	218	217	217	218	217	217	218	218	217
Knapsack	C_{max}	238	238	245	250	251	250	256	254	260	259
	# of teams	40	43	42	42	41	40	40	40	40	40
	# of idle auditors	11	1	2	0	1	16	15	14	13	11
Savings	C_{max}	226	224	224	222	225	223	224	223	222	221
	# of teams	50	50	50	50	50	50	50	50	50	50
	# of idle auditors	21	19	19	18	15	15	11	12	11	9

larger total completion time because of the large deviations of experience requirement of branches assigned to a team. On the other hand, the savings method forms more teams with varying experience levels. As expected, the larger the number of teams are formed, the better the value of C_{max} gets. The auditors that remain idle in the final solution can be appointed to teams with "high" workload.

In this paper, we present two heuristic methods for the audit scheduling problem. In the future, we will strive to improve the solutions of these heuristics using a metaheuristic algorithm, and to balance the workloads of the teams.

Acknowledgements This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) under grant 115M544.

References

1. Clarke, G., Wright, J.W.: Scheduling of vehicles from a central depot to a number of delivery points. Oper. Res. **12**, 568–581 (1964)
2. Dodin, B., Elimam, A.A., Rolland, E.: Tabu search in audit scheduling. Eur. J. Oper. Res. **106**, 373–392 (1998)
3. Drexl, A., Frahm, J., Salewski, F.: Audit-staff scheduling by column generation. In: Perspectives on Operations Research, pp. 137–162. DUV (2006)
4. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D.: Staff scheduling and rostering: a review of applications, methods and models. Eur. J. Oper. Res. **153**, 3–27 (2004)
5. Nelsen, R.B.: An introduction to copulas. Springer Science & Business Media (2007)

Machine Scheduling for Multi-product Disassembly

Franz Ehm

Abstract Increasing emergence of large amounts of discarded products puts pressure on manufacturers in the consumer goods industry. They have to balance economic considerations with environmental expectations of their costumers and legal obligations imposed by the government body. In this context, efficient disassembly helps to enable profitable remanufacturing of end-of-life products or at least limit the losses from disposal. This study represents a novel approach to combine the problems of disassembly sequence planning and machine scheduling. In contrast to existing formulations in the research field, the proposed model explicitly considers divergence of the product structure as inherent feature of disassembly. As a result, disassembly operations related to separate sub-assemblies of the same product can be scheduled at the same time. The proposed mixed integer programming formulation is solved using commercial optimization software and first computational implications are drawn from a short numerical study.

1 Introduction

Over the past three decades, varying disassembly planning problems have been addressed by operations researchers. Perhaps the most classical problem consists in determining a sequence of joint or part removal operations which satisfies all technical precedence constraints and leads to minimal disassembly cost or time, maximum revenue or profit. Several variations of the so called *disassembly sequencing problem* (DSQP) have been approached. Lambert [7] suggests a linear program using AND/OR-graphs to maximize the net revenue from complete disassembly. In contrast to completely dismounting a product into its basic components, selective disassembly problems as studied in [6] by means of a two-commodity network flow model seek to determine both the optimal level and sequence simultaneously. An extension to that model is presented by [8] using a two-stage approach to incorporate the assignment of EOL options such as reuse, recycling or remanufacturing into

F. Ehm (✉)
Industrial Management, TU Dresden, 01062 Dresden, Germany
e-mail: franz.ehm@tu-dresden.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_67

the decision making. Furthermore, numerous heuristic or meta-heuristic solution techniques have been deployed for similar problem statements [11].

In the *disassembly line balancing problem* such as studied by [5] disassembly tasks for a finite supply of single product type are sequenced and allocated to a set of linearly arranged stations in order to meet the demand of parts and a given cycle time. Interestingly however, only few publications address the interdependencies between sequence planning and time-wise resource allocation in the presence of multiple products. In the two-phase model in [3] disassembly tasks are first assigned to one of several flexible disassembly cells before forming sequences to minimize inter-cellular movements of the products. Cheng [4] propose a two-stage flow-shop formulation with dismantling taking place in the first and refurbishment being realized in the second stage. Finally, [1] address the reverse flow of products in a job-shop by combining two flow-shop problems with reversed machine order of the jobs.

Disassembly jobs may present various feasible process plans depending on product specific precedence constraints. Machine scheduling related problem statements considering process plan flexibility have emerged under the terms of *flexible job-shop scheduling* [10] or *integrated process planning and scheduling* [2]. While existing formulations are applicable to certain product types, the proposed model allows for divergence of the disassembly structure which occurs when a parent-assembly is separated into multiple child-assemblies. Since the resulting sub-assemblies can be processed independently the associated tasks are no more subject to non-overlapping constraints within the respective job.

The remainder of this work is structured as follows. In Sect. 2 we derive a graph-oriented representation of the problem based on a given product structure. Section 3 contains a comprehensive mixed integer programming (MIP) formulation. Section 4 visualizes the operation of the model using a numerical example and discusses first computational effects. Finally, Sect. 5 summarizes the findings of this paper and provides some indications for future research.

2 Graph Based Problem Representation

In a dismantling facility, a given order of n products has to be completely disassembled using a shop of m stations. Each of the products consists of a given number of parts and joints. Removal of a joint leading to the release of single components or sub-assemblies is denoted as a disassembly task k. We assume that only one station is available per task. The problem is to find a disassembly sequence for each product and a machine schedule for the corresponding tasks which minimizes makespan. For each product, feasible sequences of tasks can be represented by means of an AND/OR graph which is derived via construction of a transition matrix with respect to the connection and precedence system of the structure [7]. This work does not discuss these steps in detail but builds upon the information about feasible disassembly plans provided by existing AND/OR graphs or transition matrices. In order to be able to process this information in a combined scheduling approach we deploy

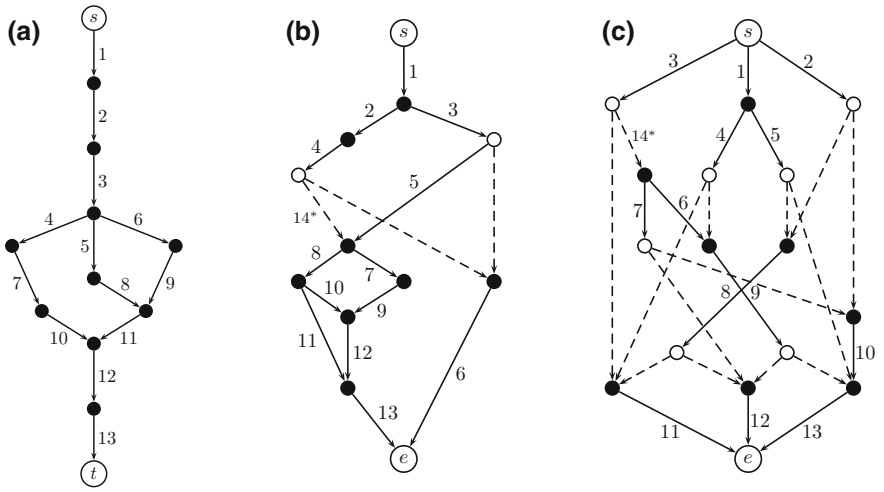


Fig. 1 Operations graphs for product 1 (**a**), 2 (**b**) and 3 (**c**): edges correspond to tasks connecting decision nodes (*solid*) or splitting nodes (*empty*)

extended operations graphs as depicted in Fig. 1. Their structure is inspired by [6] which is originally aimed at selective one-product disassembly. In this work, operations graphs are used to model precedence relations, exclusiveness and parallelism of tasks. Edges following a decision node, marked by a solid dot, denote alternative operations whereas splitting nodes, marked by empty circles, enable all outgoing edges to be passed. Additionally, dummy operations, marked as dashed edges, are introduced to avoid ambiguity when multiple edges lead to the same node. Consistently, all ingoing edges of a decision node represent alternative operations except for the sink node e. As shown in Fig. 1, disassembly may involve serial (a) or parallel execution (b, c) of tasks depending on the product structure.

3 Mixed Integer Programming Formulation

Based on existing operations graphs we formulate a MIP to determine a disassembly path for each product and schedule the corresponding operations at the stations to minimize makespan. The implementation of typical scheduling constraints are guided by the formulations of [10] and [2] which are based on the popular model of Manne [9]. The notation and the presentation of the proposed MIP are as follows:

Notation

Indices and sets

i	disassembly station index, $i \in I = \{1, 2, \ldots, m\}$
j	EOL product index, $j \in J = \{1, 2, \ldots, n\}$

k, l	disassembly operations indices, $k, l \in O_j$ for product j
P_j	Set of task pairs (k, l) of j with direct precedence "k before l"
Q_j	Set of task pairs (k, l) of j allowing parallel processing
r	Decision node index $r \in R_j$ for product j
$R_{jr}^{in}, R_{jr}^{out}$	Set of ingoing and outgoing tasks of decision node r of j
$S_{jt}^{in}, S_{jt}^{out}$	Set of ingoing and outgoing tasks of splitting node t of j
t	Splitting node index, $t \in S_j$ for product j

Parameters

I_{jk}	Singleton containing station i for processing k of j
M	Large number
p_{jk}	Processing time of operation k of product j

Decision variables

C_{max}	Makespan
c_{jk}	Integer completion time of operation k of product j
s_{jk}	Integer start time of operation k of product j
x_{ijkhl}	Boolean, 1, if l of h precedes k of j at i, 0 otherwise
y_{jkl}	Boolean, 1, if l precedes k for product j, 0 otherwise
z_{jk}	Boolean, 1, if k of j is processed, 0 otherwise

Minimize C_{max} (1)

Subject to

$$C_{max} \geq c_{jk} \qquad \forall j \in J,\ k \in O_j \tag{2}$$

$$s_{jk} \leq z_{jk} \cdot M \qquad \forall j \in J,\ k \in O_j \tag{3}$$

$$c_{jk} - s_{jk} = p_{jk} \cdot z_{jk} \qquad \forall j \in J,\ k \in O_j \tag{4}$$

$$\sum_{k \in R_{jr}^{in}} z_{jk} = \sum_{k \in R_{jr}^{out}} z_{jk} \qquad \forall j \in J,\ r \in R_j, r > 1 \tag{5}$$

$$\sum_{k \in R_{jr}^{out}} z_{jk} = 1 \qquad \forall j \in J,\ r = 1 \tag{6}$$

$$z_{jk} \geq z_{jl} \qquad \forall j \in J,\ t \in S_j,\ l \in S_{jt}^{in},\ k \in S_{jt}^{out} \tag{7}$$

$$s_{hl} - c_{jk} + M \cdot \left(1 - x_{ijkhl}\right) \geq 0 \qquad \forall j, h \in J,\ j < h,\ k \in O_j, l \in O_h,\ i \in I_{jk} \cap I_{hl} \tag{8}$$

$$s_{jk} - c_{hl} + M \cdot x_{ijkhl} \geq 0 \qquad \forall j,\ h \in J,\ j < h,\ k \in O_j,\ l \in O_h,\ i \in I_{jk} \cap I_{hl} \tag{9}$$

$$s_{jl} - c_{jk} + M \cdot \left(1 - y_{jkl}\right) \geq 0 \qquad \forall j \in J,\ k, l \in O_j | (k, l) \in Q_j, k < l, I_{jk} = I_{jl} \tag{10}$$

$$s_{jk} - c_{jl} + M \cdot y_{jkl} \geq 0 \qquad \forall j \in J,\ k, l \in O_j | (k, l) \in Q_j, k < l, I_{jk} = I_{jl} \tag{11}$$

$$s_{jl} - c_{jk} + M \cdot \left(1 - z_{jl}\right) \geq 0 \qquad \forall j \in J,\ k, l \in O_j | (k, l) \in P_j \tag{12}$$

$$c_{jk}, s_{jk} \geq 0 \qquad \forall j \in J, k \in O_j \tag{13}$$

Makespan is characterized by the latest completion time in the schedule as defined in (2). Conditions (3) and (4) ensure that solely tasks which are selected in the disassembly sequence of j are scheduled. We deploy (5) to make sure that ingoing and outgoing flow of each decision node are identical. This restriction is supplemented by (6) which specifies that exactly one task option is chosen at the start node $r = 1$. (7) enforces that all outgoing operations of a splitting node are realized providing that this node is reached via a selected task. Referring to [2] and [10] we use (8), (9) and (10), (11) to implement non-overlapping constraints for each station and within each job, respectively. In contrast to these formulations, we assume that only one station is available for each task and thus omit the machine index in the start and completion times. Furthermore, non-overlapping within a job is relaxed for combinations of tasks that belong to disjoint sub-assemblies enabling parallel processing. Finally, (12) ensures that task l can only be started after any of its predecessors k has been completed providing that l is selected in the disassembly sequence of j.

4 Numerical Study

The operation of the model is illustrated using a small example of three different EOL products and three disassembly stations. The products are represented by the operations graphs depicted in Fig. 1. Data for station assignment and processing times is set arbitrarily. The model is solved within less than 1 sec using standard optimization software CPLEX and running an Intel® i5 CPU@3.40 GHz and 4 GB memory. As shown in the Gantt chart in Fig. 2 divergent product structures 2 and 3 enable overlapping of some of the tasks in the schedule. Notably, tasks 10 and 12 of product 3 and operations 6 and 11 of job 2 can be processed in parallel. Note that dummy operations do not appear in the schedule due to processing times of zero.

Preliminary testing was conducted considering 9 problem configurations for three different product types and solving samples of 10 random instances each. It revealed a rapidly increasing solution time with larger values of n and smaller m as illustrated in Fig. 3. On the one hand, computational effort is driven by the amount of real and dummy operations in the problem which is essentially controlled by the number of jobs. On the other hand, involving more disassembly stations reduces the number of tasks sharing the same station and thus results in fewer non-overlapping constraints in the model. Product structure is another important aspect. Providing identical data

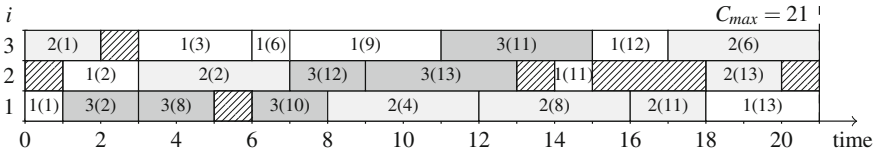


Fig. 2 Gantt chart for the optimal solution of exemplary three-product disassembly

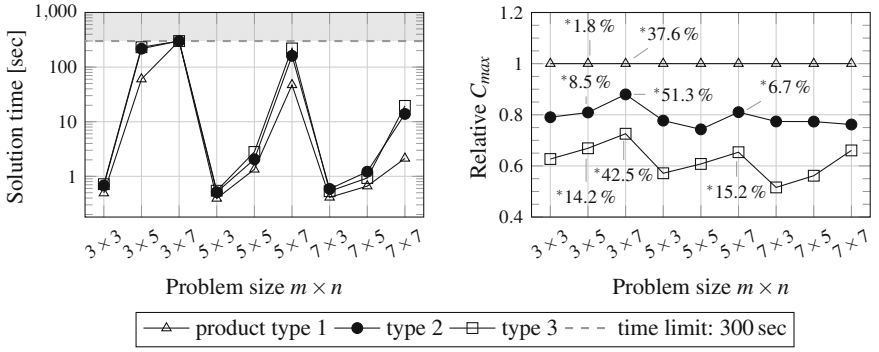


Fig. 3 Solution time and relative objective value (* average MIP gap) from samples of 10 random instances with varying number of stations m and jobs n assuming the product types shown in Fig. 1

for the number of tasks per job, processing times and station assignment across all three structures we observe higher degrees of parallelism in disassembly to result in shorter schedules while increasing computational effort.

5 Conclusions

In this paper, a novel approach to integrate the problems of disassembly sequencing and machine scheduling for multiple products was presented. Based on product structure information a graph-oriented representation was introduced to facilitate application of existing scheduling formulations. We then developed a MIP model which was illustrated by a numerical example and tested on 270 random instances. However, more extensive testing has to be undertaken in order to better assess the performance of the model. In this context, feasible random product graphs should be automatically generated and used instead of arbitrary disassembly structures.

References

1. Abdeljaouad, M.A., Bahroun, Z., Omrane, A., Fondrevelle, J.: Job-shop production scheduling with reverse flows. Euro. J. Oper. Res. **244**(1), 117–128 (2015)
2. Amin-Naseri, M., Afshari, A.J.: A hybrid genetic algorithm for integrated process planning and scheduling problem with precedence constraints. Int. J. Adv. Manuf. Technol. **59**(1–4), 273–287 (2012)
3. Andrés, C., Lozano, S., Adenso-Diaz, B.: Disassembly sequence planning in a disassembly cell context. Robot. Comput.-Integr. Manuf. **23**(6), 690–695 (2007)
4. Cheng, T., Lin, B., Tian, Y.: A scheduling model for the refurbishing process in recycling management. Int. J. Product. Res. **51**(23–24), 7120–7139 (2013)

5. Kalaycılar, E.G., Azizoğlu, M., Yeralan, S.: A disassembly line balancing problem with fixed number of workstations. Euro. J. Oper. Res. **249**(2), 592–604 (2016)
6. Kang, J.-G., Lee, D.-H., Xirouchakis, P., Persson, J.-G.: Parallel disassembly sequencing with sequence-dependent operation times. CIRP Ann.-Manuf. Technol. **50**(1), 343–346 (2001)
7. Lambert, A.J.: Determining optimum disassembly sequences in electronic equipment. Comput. & Ind. Eng. **43**(3), 553–575 (2002)
8. Ma, Y.-S., Jun, H.-B., Kim, H.-W., Lee, D.-H.: Disassembly process planning algorithms for end-of-life product recovery and environmentally conscious disposal. Int. J. Prod. Res. **49**(23), 7007–7027 (2011)
9. Manne, A.S.: On the job-shop scheduling problem. Oper. Res. **8**(2), 219–223 (1960)
10. Özgüven, C., Özbakır, L., Yavuz, Y.: Mathematical models for job-shop scheduling problems with routing and process plan flexibility. Appl. Math. Model. **34**(6), 1539–1548 (2010)
11. Ullerich, C.: Advanced Disassembly Planning/Flexible, Price-quantity Dependent, and Multi-period Planning Approaches. Springer Gabler, Wiesbaden (2014)

A Hybrid Metaheuristic for the Multi-mode Resource Investment Problem with Tardiness Penalty

Patrick Gerhards and Christian Stürck

Abstract In this work we propose and analyze a hybrid approach for the multi-mode resource investment problem with tardiness penalty (MRIPT). The MRIPT is a project scheduling problem where, for a given deadline, the objective is to minimize the costs of resources allocated to the project as well as tardiness penalty costs for not respecting the given deadline. For each project activity multiple execution modes with differing resource requirements and durations are given. In particular, we propose a large neighborhood search where destroy operators are applied to a feasible solution to obtain subproblems. These subproblems are solved with MIP-based recreate operators to obtain an improved solution.

1 Introduction

The resource investment problem (RIP, also called resource allocation cost problem (RACP)) was first considered by Möhring [4]. In contrast to resource constrained project scheduling problems (RCPSP) he describes them as the "problem of scarce time". For a given project deadline the objective is to find a resource allocation that minimizes the resource costs. Compared to the resource constrained project scheduling problem and its multi-mode extension (MRCPSP) that have been extensively studied, the resource investment problem received relatively little attention. For the RIP several exact (e.g. [1, 6]) and metaheuristic approaches (e.g. [5, 9]) have been studied. Hsu and Kim [2] considered a multi-mode extension of the RIP and Shadrokh and Kianfar [7] introduced an extension of the RIP where exceeding the due date is permitted but penalized with tardiness costs (RIPT). To our knowledge, an extension of the RIP with multiple modes and tardiness penalty costs has received very little attention by the scientific community so far. In this work we pro-

P. Gerhards (✉) · C. Stürck
Helmut Schmidt University, Hamburg, Germany
e-mail: patrick.gerhards@hsu-hh.de

C. Stürck
e-mail: christian.stuerck@hsu-hh.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_68

pose a hybrid metaheuristic for the multi-mode resource investment problem with tardiness penalty (MRIPT) and evaluate it on modified instances for the MRCPSP.

This article is structured as follows: In Sect. 2 we will give a formal definition of the MRIPT and a mathematical model. In Sect. 3 the proposed hybrid metaheuristic is presented and in Sect. 4 we present selected computational experiments and give an outlook on further research. With this work we want to show that this type of hybrid metaheuristic can be effectively applied to this problem.

2 Problem Formulation

The MRIPT is defined by the following properties: A set of nonpreemptable activities $A = \{0, \ldots, n+1\}$, precedence constraints $E = \{(i,j) : i,j \in A\}$, a set $\mathscr{R}$ of renewable resources and a set $\mathscr{R}^n$ of nonrenewable resources. For each activity i there is a set M_i of modes that can be chosen for the execution of activity i. If mode $m \in M_i$ is chosen, activity i has duration $d_{i,m} \in Z^+$ and it has a resource consumption $r_{i,m,k} \in Z^+$ for each resource $k \in \mathscr{R} \cup \mathscr{R}^n$. A due date $D \in Z^+$ for the makespan of the project and tardiness penalty cost factor $c_t \in Z^+$ are given. For each resource $k \in \mathscr{R} \cup \mathscr{R}^n$ the available capacity of the resource has to be chosen and resource cost factors $c_k \in Z^+$ are given. The objective is to find a precedence and resource feasible schedule that minimizes the sum of tardiness and resource costs.

$$\min \sum_{t=D+1}^{LS_{n+1}} c_t \cdot (x_{n+1,1,t} \cdot (t-D)) + \sum_{k \in \mathscr{R} \cup \mathscr{R}^n} c_k \cdot a_k \tag{1}$$

$$s.t. \sum_{m \in M_i} \sum_{t=ES_i}^{LS_i} x_{i,m,t} = 1 \quad \forall i \in A \tag{2}$$

$$\sum_{m \in M_i} \sum_{t=ES_i}^{LS_i} x_{i,m,t} \cdot (t + d_{i,m}) \leq \sum_{m \in M_j} \sum_{t=ES_j}^{LS_j} x_{j,m,t} \cdot t \quad \forall (i,j) \in E \tag{3}$$

$$\sum_{i \in A} \sum_{m \in M_i} \sum_{t=ES_i}^{LS_i} x_{i,m,t} \cdot r_{i,m,k} \leq a_k \quad \forall k \in \mathscr{R}^n \tag{4}$$

$$\sum_{i \in A} \sum_{m \in M_i} \sum_{q=\max(ES_i, t-d_{i,m})}^{\min(t, LS_i)} x_{i,m,q} \cdot r_{i,m,k} \leq a_k \quad \forall k \in \mathscr{R}, \forall t \in T \tag{5}$$

$$x_{i,m,t} \in \{0,1\} \quad \forall i \in A, \forall m \in M_i, t = ES_i, \ldots, LS_i \tag{6}$$

$$a_k \in Z^+ \quad \forall k \in \mathscr{R} \cup \mathscr{R}^n \tag{7}$$

In the mathematical model we define binary decision variables $x_{i,m,t}$ which are set to 1 if and only if activity i starts in mode m in period t (see (6)) and integer-valued decision variables a_k which represent the available capacity of resource k (see (7)).

For each activity i we calculate a lower bound ES_i and an upper bound LS_i for its possible starting period using the critical path methods (CPM).

The objective function (1) minimizes the sum of tardiness costs and resource costs. Equation (2) makes sure that for every activity i exactly one mode and one starting time is assigned. With constraint (3) we ensure the precedence constraints. Constraints (4) and (5) model the nonrenewable and renewable resource requirements, respectively.

3 Proposed Hybrid Metaheuristic

The proposed hybrid metaheuristic consists of a large neighborhood search (LNS) which is used as a master search algorithm. The concept of LNS was first applied to vehicle routing problems by Shaw [8]. Given a feasible solution, in each iteration of the LNS large parts of the current solution are destroyed and then those destroyed parts are recreated to a (hopefully better) solution. Several destroy and recreate operators can be used.

Our proposed approach is both a hybrid metaheuristic and a matheuristic since it uses mathematical programming in the recreate step. We define a solution *candidate* to consist of a vector of start and finish periods for all activities, a vector of modes for all activities and resource availabilities for all resources. A *destroyed candidate* consists of earliest and latest start times (EST and LST, respectively) for each activity and a list of executable modes for each activity.

The LNS algorithm is displayed in Fig. 1. In line 3 a destroy operator d is chosen according to the probability vector P. These probabilities can vary based on the properties of the problem instance. The call of the destroy operator in line 4 returns a destroyed candidate *destroyedC*. Here, *freeVar* is an upper bound on the number of decision variables in the MIP that is solved in the recreate step. In line 5 the recreate function r is called to obtain a solution candidate $c^{proposal}$. The recreate function

Data: initial candidate $c^{initial}$, pool of destroy operators D, probabilities P, number free variables $freeVar$
Result: best obtained candidate c^{best}

```
1 c^best := c^initial
2 while stopping criterion is not met do
3 |   Choose a destroy operator d from D with probabilities in P
4 |   destroyedC := d(c^best, freeVar)
5 |   c^proposal := r(c^best, destroyedC)
6 |   if costs(c^proposal) < costs(c^best) then
7 |   |   c^best := c^proposal
8 return c^best
```

Fig. 1 LNS

solves a MIP using the mathematical model presented in Sect. 2 but with the EST, LST and list of modes stored in *destroyedC*. Hence, it is possible to "fix" some of the activities by setting their EST and LST in the destroyed candidate equal to the start time in c^{best}. Additionally, we can add only the mode selected in c^{best} which speeds up the solving of the MIP a lot. Only if a better solution candidate was found, c^{best} is updated in line 7.

As a stopping criterion we use the number of evaluated schedules which is common practice in project scheduling. Since we use MIP techniques we define the number of evaluated schedules by one call of the recreate operator as the number of decision variables of the activities in the MIP divided by the number of activities. For example, if we allowed only one mode for each activity and set the EST and LST to be equal, that would result in one evaluated schedule.

The first destroy operator is called *destroyHighResourcePeriods* and works well if the tardiness penalty costs c_t are relatively low compared to the resource costs. It aims to find activities that are scheduled (in the current candidate) in time periods where the usage of renewable resources is above the average renewable resource usage per time period. While the threshold *freeVar* of decision variables is not exceeded, *destroyHighResourcePeriods* randomly selects an activity that is scheduled in a "high resource usage period" and sets its EST and LST to the values calculated by CPM. Then, all available modes are added to the list in the destroyed candidate. For example, if an activity with $EST = 10, LST = 19$ and 3 modes is chosen, it would result in 30 decision variables for that activity in the MIP. For all activities that are not chosen, i.e. they are not scheduled in high resource usage periods or the threshold was exceeded, we add only the mode that was used in the current solution candidate to the mode list in the destroyed candidate and set the EST in the destroyed candidate to be the starting period in the current solution. However, we set the LST in the destroyed candidate to be the LST computed by CPM. This increases the number of decision variables in the MIP slightly but also allows to schedule these activities at a later period.

The second destroy operator is called *destroyLowResourcePeriods* and does the opposite of the first operator. Here, we search for activities that are scheduled in time periods with a renewable resource usage below the average renewable resource usage per time period. Similarly, we set the EST and LST of randomly selected activities that are scheduled (in the current candidate) in such areas to those computed by CPM and add all their modes in the destroyed candidate. For the unchosen activities we allow only the mode used in the current candidate and earlier start times, i.e. we set the LST to the start period of the current candidate and the EST to the EST computed by CPM. This operator is expected to work well if the tardiness costs are relatively high compared to resource costs.

The initial candidate is generated simply by assigning the first mode to each activity and assigning a schedule with a serial generation scheme where every activity has equal priority. The resource capacities are then set to values such that the initial candidate is feasible.

4 Computational Experiments and Outlook

We implemented a software prototype of the proposed LNS in C# and used CPLEX 12.6.3 as a solver for the MIP in the recreation step. Since to our knowledge no established benchmark instances for the MRIPT exist, we used the J30 instances of the PSPLIB [3] for the MRCPSP and modified them. In J30 there are 640 instances with projects consisting of 30 activities and 3 modes per activity. They each have 2 renewable and 2 non-renewable resources. To get MRIPT instances we draw numbers from a discrete uniform distribution $\mathscr{U}\{1,5\}$ for the resource cost factors and one random number from a discrete uniform distribution $\mathscr{U}\{1,10\}$ for the tardiness penalty cost factor for each instance in J30. Based on these cost values, for each instance in J30 we create six new MRIPT instances that only differ in the due date. As a base value for the due date we take the earliest finishing time of the project (obtained by CPM) and increase it by 0%, 10%, … , 50% for each instance type. This results in six classes of instances that are named MRIPJ30_1 , … , MRIPJ30_6, each consisting of 640 instances. For all instances we computed a lower bound (LB) using the linear programming relaxation of the MIP.

Our first experiment was conducted only on MRIPJ30_1 and tested the behaviour of the destroy operators as well as the influence of the parameter *freeVar*. In Table 1, LNS (LowResourcePeriods) used only *destroyLowResourcePeriods* while LNS (HighResourcePeriods) exclusively applied *destroyHighResourcePeriods* as a destroy operator. In the case of LNS (dynamic) both destroy operators are used and the probabilities of their usage depend on the cost values of the instance (for instances with relatively low tardiness cost factor *destroyHighResourcePeriods* was used two times more often and vice versa).

We see that a higher value of *freeVar* improves the results which was expected since the number of decision variables in the MIP defines the size of the neighborhood that is searched. The destroy operator *destroyLowResourcePeriods* performs better than *destroyHighResourcePeriods* when they are solely used but the combination of the two operators outperforms both. Hence, we decided for further experiments to present only results for the dynamic variant that uses both operators.

In Table 2 we show the results of LNS (dynamic) for each instance class after 1,000 and 5,000 schedules and the parameter *freeVar* = 2,000. For the rather poor lower bounds retrieved by the LP relaxation the proposed procedure achieves good

Table 1 Results on MRIPJ30_1 for 5,000 evaluated schedules

Method	*freeVar*	Avg deviation LB (%)	Avg time in s
LNS (LowResourcePeriods)	500	9.95	1,709
LNS (HighResourcePeriods)	500	29.31	762
LNS (dynamic)	500	6.56	1,521
LNS (LowResourcePeriods)	2,000	8.53	1,321
LNS (HighResourcePeriods)	2,000	20.54	748
LNS (dynamic)	2,000	6.22	1,250

Table 2 Results on all MRIPJ30 instances with LNS (dynamic) and *freeVar* = 2,000

Number of schedules	1,000		5,000	
Instances	Avg deviation LB (%)	Avg time in s	Avg deviation LB (%)	Avg time in s
MRIPJ30_1	6.57	258	6.23	1,246
MRIPJ30_2	6.83	260	6.47	1,243
MRIPJ30_3	7.06	258	6.69	1,232
MRIPJ30_4	7.36	253	6.91	1,208
MRIPJ30_5	7.46	244	6.99	1,151
MRIPJ30_6	7.46	234	7.00	1,093

results but it seems to work better on instances with smaller due dates. It is interesting that the evaluation of more schedules increases the quality of the solutions only slightly. Therefore, we need to do further investigation on how to control the neighborhood size with *freeVar* during the search. Since in many iterations of the LNS no better solution is found we also want to focus on improving the destroy operators. In further research, the parameter *freeVar* should be adapted during the search but we still need to identify the best strategy for this. Since the results on these instances with 30 activities are satisfactory we will also test the procedure on instances with 50 or 100 activities.

References

1. Demeulemeester, E.: Minimizing resource availability costs in time-limited project networks. Manag. Sci. **41**(10), 1590–1598 (1995)
2. Hsu, C.C., Kim, D.S.: A new heuristic for the multi-mode resource investment problem. J. Oper. Res. Soc. **56**(4), 406–413 (2005)
3. Kolisch, R., Sprecher, A.: PSPLIB—a project scheduling problem library. Eur. J. Oper. Res. **96**(1), 205–216 (1997)
4. Möhring, R.H.: Minimizing costs of resource requirements in project networks subject to a fixed completion time. Oper. Res. **32**(1), 89–120 (1984)
5. Ranjbar, M., Kianfar, F., Shadrokh, S.: Solving the resource availability cost problem in project scheduling by path relinking and genetic algorithm. Appl. Math. Comput. **196**(2), 879–888 (2008)
6. Rodrigues, S.B., Yamashita, D.S.: An exact algorithm for minimizing resource availability costs in project scheduling. Eur. J. Oper. Res. **206**(3), 562–568 (2010)
7. Shadrokh, S., Kianfar, F.: A genetic algorithm for resource investment project scheduling problem, tardiness permitted with penalty. Eur. J. Oper. Res. **181**(1), 86–101 (2007)
8. Shaw, P.: Using constraint programming and local search methods to solve vehicle routing problems. In: International Conference on Principles and Practice of Constraint Programming, pp. 417–431. Springer (1998)
9. Van Peteghem, V., Vanhoucke, M.: An artificial immune system algorithm for the resource availability cost problem. Flex. Serv. Manuf. J. **25**(1–2), 122–144 (2013)

A Decomposition Method for the Multi-Mode Resource-Constrained Multi-Project Scheduling Problem (MRCMPSP)

Mathias Kühn, Sebastian Dirkmann, Michael Völker and Thorsten Schmidt

Abstract Multi-Mode Resource-Constrained Multi-Project Scheduling Problems (MRCMPSP) with large solution search spaces cannot be optimized in an acceptable computation time. In this paper, we have focused on decomposition strategies for such large scale problems. Based on literature review a time-based decomposition approach was adopted for the present problem. With time-based decomposition approaches a schedule is divided into several time periods. All activities in a time period describe an independent problem, termed as a sub-problem. Due to the independent optimization of these sub-problems project information regarding the relationships among activities in different time periods is not considered. This loss of information has a negative impact on the overall solution quality. We developed a decomposition strategy to improve the interactions between the sub-problems for a better target performance while reducing the computation time. Based on an initial solution the sub-problems are created and sequentially optimized in a concept similar to rolling horizon heuristics. We introduce a transition stage with a constant and a variable component at the end of each partial schedule to improve the interactions among sub-problems and thus taking the volatile nature of the examined problems into account. In comparison, our approach proved to provide significant improvements in runtime and target performance.

1 Introduction

Currently large scale Multi-Mode Resource-Constrained Multi-Project Scheduling Problems (MRCMPSP) cannot be optimized in an acceptable computation time. Previous experiments showed that the computation time needed to generate sufficient results increases exponentially with larger problem size. (Fig. 1) We present an approach to significantly reduce the computation time for large MRCMPSPs.

M. Kühn (✉) · S. Dirkmann · M. Völker · T. Schmidt
Institut Für Technische Logistik Und Arbeitssysteme, Dresden University of Technology, Dresden, Germany
e-mail: mathias.kuehn@tu-dresden.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_69

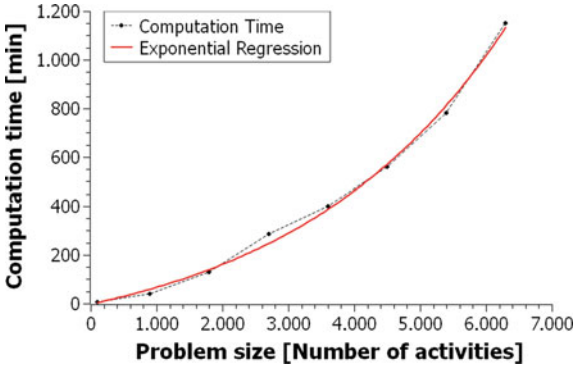


Fig. 1 Experimental comparison between computation time and problem size

This paper is organized as follows. Section 2 provides a brief description of the problem. The decomposition method is introduced in Sect. 3. Section 4 presents the computational experiments and their results, while Sect. 5 provides a summary with suggestions for future research in the proposed research area.

2 Problem Description

The Resource-Constrained Scheduling-Problem (RCPSP) has been extensively used in practical applications. Nevertheless, the model does not incorporate all aspects of real-world problems [8]. Therefore many extensions of the RCSPSP have been presented [4]. The MRCMPSP generalizes the RCPSP in two ways. Activities can be executed in multiple modes (MRCPSP) and multiple projects have to simultaneously compete for the same resources (RCMPSP) (Fig. 2). The RCPSP and thus, in extension, MRCMPSP have been acknowledged as NP-hard [2]. These kind of schedul-

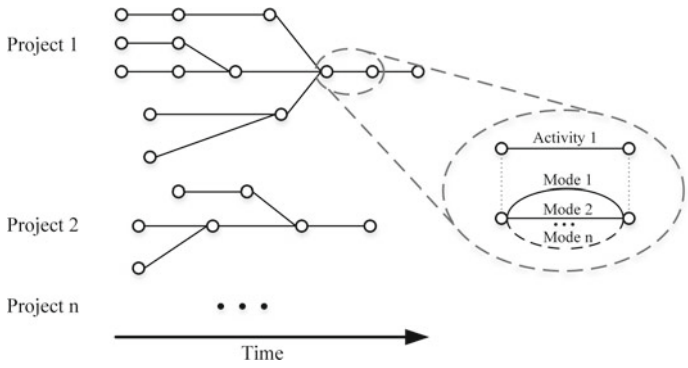


Fig. 2 The multi-mode resource-constrained multi-project scheduling problem

ing problems are commonly solved by heuristics [5]. In this paper a metaheuristic implemented in a simulation-based optimization platform is used [1]. Besides the improvement in heuristic approaches and the adoption of machine learning techniques, decomposition has experienced a lot of scientific attention.

Decomposition methods attempt to decompose complex problems into sub-problems, which are easier to optimize and understand. Solutions are generated for each sub-problem individually and integrated to solve the initial complete problem.

Sub-problems can be created by decomposition or aggregation along two main axes—time and scheduling entities (e.g. resources) [6]. A systematic literature review on the MRCMPSP and other relevant problem indicates that temporal and hierarchical approaches have the biggest potential regarding reduction of complexity and adaptability. Although, two decomposition methods have been proposed in literature, none of them has been used in an environment based on simulation based optimization [3, 7]. Due to the large order and project sizes of the examined problems a hierarchical decomposition is not able to sufficiently simplify the problem. A flexible time based approach, which is easier to implement, is chosen instead. Deficit of current time based approaches is the negligence of relationships among activities in different time periods during their independent optimization which results in a deterioration of the overall solution quality. Our approach tries to compensate for this shortcoming.

3 Decomposition Method

The proposed decomposition procedure as depicted in Fig. 3 operates as follows: The initial base solution is converted into several sub-problems by dividing it into time intervals. The time intervals for generation of sub-problems are created by dividing

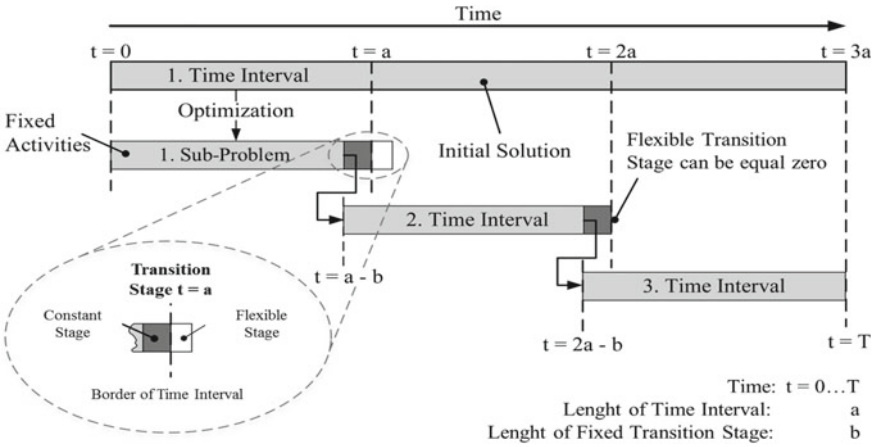


Fig. 3 Decomposition procedure

the base solution into equal parts. Basically, a sub-problem represents parts of baseline schedule where a string of tasks has been assigned resources and the objective is to optimize the user defined criteria. Since the sub-problems are determined without taking into account the project structure, a certain risk regarding coherence of projects is involved. For instance, it is possible that for a certain assembly group, only one task is assigned to the current sub-problem and the rest of that particular group has been assigned to the following sub-problem. Since this activity can be shifted during the optimization far away from the rest of the activities and therefore, total cycle time for this assembly group increases. The reason behind this problem is the missing successor information. To avoid that lack of information, we developed a strategy which consists of components based on a constant and flexible amount of activities at the transitions stages. The constant component takes a fixed amount of activities from the optimized sub-problem to the following sub-problem while the flexible component includes all activities which are started after the previously defined time interval. The flexible component is optional and results from the optimization. Based on the exponential growth in computation time, a smaller problem size consequently leads to a reduction in computation time. The separate solution process of time intervals also enables the usage of individual strategies depended on the properties of the sub-problem. We are currently working on the extension of the method by considering different assignment scenarios for the constant activities and also to add a constant component that can provide information about the predecessors from the previous sub-problem.

4 Computational Experiments

To analyze the performance of the decomposition method computational experiments were performed. We designed a number of experiments to evaluate the correlation between the base schedule and the number of time intervals regarding the overall optimization results as well as a comparison of the computational runtime. A Design of Experiments is shown in Table 1. The input parameters for the base schedules are set in four steps which determine their overall solution quality. For every step four different time interval sizes are compared. For each experiment the best sched-

Table 1 Experiment design

Experiments	Base schedule		
	Generation size	Population size	Number of intervals
1-4	1	1	1 2 3 4
5-8	5	10	1 2 3 4
9-12	25	50	1 2 3 4
13-16	50	100	1 2 3 4

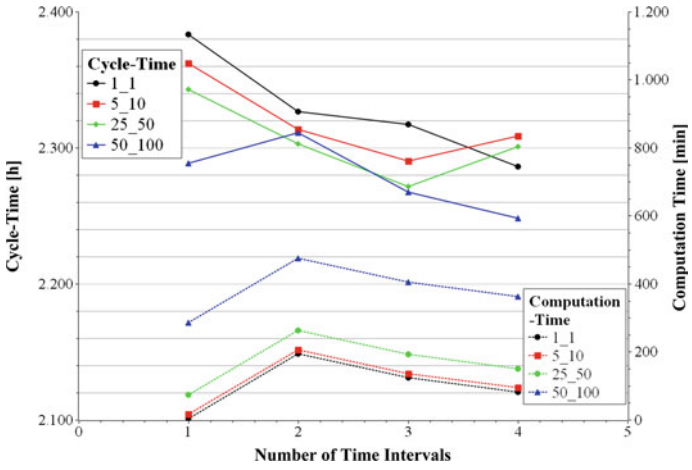


Fig. 4 Comparison between cycle-time, computation time and number of time intervals

ule with respect to the objective cycle time is selected for further investigations. Sub-problems are optimized with generation size 50 and population size 100. The constant transition stage was defined as 10% of the activities per time interval. The results are depicted in Fig. 4. An analysis of the experiments implies that cycle-time improves with increasing quality of base schedules. While the improvement of cycle-time in the conducted experiments was observed to be linear, the run time decreases approximately exponentially with a higher number of time intervals. The cycle-time for experiments with the same base schedule shows significant improvement with a higher quantity of time intervals.

The improvement between each individual step is nearly constant. Furthermore, the computation time, which is calculated by the accumulation of the time needed to generate the base schedule and optimize each interval, shows an increase until two sub-problems and decreases afterwards. This can be affiliated to the correlation between computation time and model size. Hence, with an increase in the number of intervals, the total computation time approaches the time needed for generating the base schedule. We were able to surpass the best solution of the conventional optimization 50_100 with a cycle-time of 2288 h and computation time of 285 min, with a base schedule 25_50 and three time intervals in a computation time of 193 min. The best a cycle-time of 2248 h was reached with the highest quality base schedule and four time intervals. The optimal computation time for the decomposition method of 82 min was also able to improve the target value of the best conventional solution. Considering the average cycle-time for each quantity of time intervals, three or four sub-problems are an improvement in comparison to the best conventional solution. Overall, we were able to achieve a significant improvement of the target value and and computation time based on the developed decomposition method.

5 Conclusions

In this paper, we investigated decomposition methods for the MRCMPSP. We came to the conclusion that there is a lack of application-oriented methods for discrete event decomposition. Especially the interactions between sub-problems was neglected in previous research of this field. Therefore we developed a strategy for optimizing the performance of the decomposition according to the objective value by using flexible transitions between the sub-problems. Several experiments were carried out to analyze and evaluate the best compromise between the quality of the base schedule and the amount of decomposition intervals concerning the computation time. Our conclusion is that the proposed decomposition method provides good results, even for low quality base schedules. The optimal parameter for generating the base schedule and the size of the time intervals still needs further investigation. In future, we will focus on performing such experiments with different input models, incorporating varying structure.

Acknowledgements The presented work is a result of the research project Simulationsbasierte dynamische Heuristik zur verteilten Optimierung komplexer Mehrziel-Multiprojekt-Multiressourcen-Produktionsprozesse (Founded by the Deutsche Forschungsgemeinschaft (DFG), Duration 04/2014-03/2017). The authors would like to thank Evangelos Angelidis and Daniel Bohn for providing the software SBOP and the constructive discussions.

References

1. Angelidis, E., Bohn, D., Rose, O., Carl, S.: Simulation-based optimization for complex assembly lines with workforce constraints. Simulation in Produktion und Logistik (2013)
2. Blazewicz, J., Lenstra, J., Kan, A.: Scheduling subject to resource constraints: classification and complexity. Discrete Appl. Math. **5**(1), 11–24 (1983)
3. Can, A., Ulusoy, G.: Multi-project scheduling with two-stage decomposition. Ann. Oper. Res. **217**(1), 95–116 (2014)
4. Hartmann, S., Briskorn, D.: A survey of variants and extensions of the resource-constrained project scheduling problem. Eur. J. Oper. Res. **207**, 1–14 (2010)
5. Kolisch, R., Padman, R.: An integrated survey of deterministic project scheduling. Omega **29**(3), 249–272 (2001)
6. Ovacik, I.M., Uzsoy, R.: Decomposition Methods for Complex Factory Scheduling Problems. Springer Science & Business Media (1997)
7. Toffolo, T., Santos, H.G., Carvalho, M.A.M., Soares, J.A.: An integer programming approach to the multimode resource-constrained multiproject scheduling problem. J. Sched. **19**(3), 295–307 (2016)
8. Wauters, T., Kinable, J., Smet, P., Vancroonenburg, W., Berghe, G.V., Verstichel, J.: The multi-mode resource-constrained multi-project scheduling problem. J. Sched. **19**(3), 271–283 (2016)

Lower Bounds for the Two-Machine Flow Shop Problem with Time Delays

Mohamed Amine Mkadem, Aziz Moukrim and Mehdi Serairi

Abstract We consider the flow shop problem with two machines and time delays with respect to the makespan, i.e., the maximum completion time. We recall the lower bounds of the literature and we propose new relaxation schemes. Moreover, we investigate a linear programming-based lower bound that includes the implementation of a new dominance rule and a valid inequality. A computational study that was carried out on a set of 480 instances including new hard ones shows that our new relaxation schemes outperform the state of the art lower bounds.

1 Introduction

This paper is devoted to dealing with the flow shop scheduling problem with two machines and time delays, denoted by $F2|l_j|C_{max}$. Let $I = (J, p_1, l, p_2)$ be an instance of $F2|l_j|C_{max}$, where $J = \{1, 2, \ldots, n\}$ is a set of n jobs, p_1 and p_2 are the vectors of processing times on the first and the second machines, and l is the vector of the time delays. Each job j has two operations. The first operation (resp. the second operation) must be executed without preemption during $p_{1,j}$ (resp. $p_{2,j}$) time units on M_1 (resp. M_2). For each job $j \in J$, a time delay of at minimum l_j time units must separate the end of the first operation and the start of the second one. The objective is to find a feasible schedule that minimizes the completion time of the last scheduled job on M_2. A feasible schedule is such that at most one operation is processed at a time on a given machine. In addition, the operations are executed without preemption, where interruption and switching of operations are not allowed.

M.A. Mkadem (✉) · A. Moukrim · M. Serairi
Sorbonne universités, Université de Technologie de Compiégne, CNRS,
UMR 7253 Heudiasyc, CS 60319, 60203 Compiégne Cedex, France
e-mail: mohamed-amine.mkadem@hds.utc.fr

A. Moukrim
e-mail: aziz.moukrim@hds.utc.fr

M. Serairi
e-mail: mehdi.serairi@hds.utc.fr

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_70

Mitten [2] proves that the permutation flow shop $F2\pi|l_j|C_{max}$, where a feasible schedule consists in having the same job sequence on both machines, can be solved in polynomial time. However, solving our problem as a permutation flow shop does not necessarily provide an optimal solution. $F2|l_j|C_{max}$ is an NP-hard problem in the strong sense even with unit-time operations [3].

The objective of this paper is to introduce new lower bounds. First, we improve the most promising lower bound of the literature. Second, we investigate a linear programming-based lower bound.

2 Combinatorial Lower Bounds

We present here the lower bounds of the literature and propose new ones. Hereafter, $C^*_{max}(I)$ represents the optimal makespan value of instance I and $C_{max}(S)$ stands for the makespan value of schedule S.

First, we survey four lower bounds of Yu [3]. We start by two $O(n)$ basic lower bounds $LB_1 = \max_{1\leq j\leq n}(p_{1,j} + l_j + p_{2,j})$ and $LB_2 = \max(\sum_{j=1}^{n} p_{1,j} + \min_{1\leq j\leq n}(l_j + p_{2,j}), \sum_{j=1}^{n} p_{2,j} + \min_{1\leq j\leq n}(l_j + p_{1,j}))$. Moreover, Yu [3] interested in the problem where each job $j \in J$ is splitted into $\min(p_{1,j}, p_{2,j})$ unitary sub-jobs. The lower bound $LB_3 = \lceil(\sum_{j=1}^{n} \min(p_{1,j}, p_{2,j}).l^u_j)/\sum_{j=1}^{n} \min(p_{1,j}, p_{2,j})\rceil + 1 + \sum_{j=1}^{n} \min(p_{1,j}, p_{2,j})$ was introduced, where $l^u_j = l_j + \max(p_{1,j}, p_{2,j}) - 1$ is the time delay observed by each sub-job derived from $j \in J$.

The fourth lower bound is presented as follows. Let S^* be an optimal schedule and $p_{k,[\ell]}$ the processing time of the job scheduled at position ℓ on M_k, $k \in \{1, 2\}$. Moreover, let j^k be the position of job j on M_k, $k \in \{1, 2\}$. For each job $j \in J$, it holds that $C_{max}(S^*) \geq \sum_{\ell=1}^{j^1} p_{1,[\ell]} + l_j + \sum_{\ell=j^2}^{n} p_{2,[\ell]}$. By adding together the above equations for all jobs and by considering that the makespan is integral, $LB_4 = \lceil(\sum_{j=1}^{n} l_j + \sum_{m=1}^{n} \rho_{1,m} + \sum_{m=1}^{n} \rho_{2,m})/n\rceil$ is a valid lower bound, where $\rho_{k,m}$ is the sum of the m smallest values in $\{p_{k,1}, p_{k,2}, \ldots, p_{k,n}\}$.

The following lower bounds were introduced by Dell'Amico [1]. In the first one, it is assumed that all jobs are executed at time 0 on M_1. The problem is then a single-machine scheduling problem with release dates denoted by $1|r_j|C_{max}$. Let I_r be the instance for $1|r_j|C_{max}$ with $r_j = p_{1,j} + l_j$ and $p_j = p_{2,j}$, $j \in J$. Obviously, $L_1 = C^*_{max}(I_r)$ is a valid lower bound on the $F2|l_j|C_{max}$ original instance, which can be computed in $O(n \log n)$-time by scheduling the jobs in a nondecreasing order of $r_j, j \in J$. By interchanging the role of M_1 and M_2, we yield a symmetric lower bound called L_2. Finally, we define the lower bound $LB_5 = \max(L_1, L_2)$.

Solving our problem as a permutation flow shop does not necessarily provide an optimal solution. However, special cases exist where it is true. Dell'Amico [1] proved that permutation schedules are dominant if $l_j \leq \min_{1\leq i\leq n}(p_{1,i} + l_i)$, $j \in J$ and then he introduced the following lower bound. Let $\bar{I} = (J, p_1, \bar{l}, p_2)$ be a new instance that is derived from instance $I = (J, p_1, l, p_2)$, where $\bar{l}_j = \min(l_j, \min_{1\leq i\leq n}(l_i + p_{1,i}))$, $j \in J$. Since $\bar{I}$ verifies Dell'Amico's [1] conditions, an optimal solution for $\bar{I}$ can be found

in polynomial time using Mitten algorithm [2]. Therefore, $LB_6 = C^*_{max}(\bar{I})$ is a valid lower bound.

Furthermore, we introduce two new lower bounds which can be considered as a generalization of LB_6. In fact, Yu [3] extended Dell'Amico's [1] result after showing that the permutation schedules are dominant if $l_j \leq \min_{1\leq i\leq n}(l_i + \max(p_{1,i}, p_{2,i}))$, $j \in J$. Therefore, from an instance $I = (J, p_1, l, p_2)$ of $F2|l_j|C_{max}$ problem, we derive a new instance $\tilde{I}(J, p_1, \tilde{l}, p_2)$, where $\tilde{l}_j = \min(l_j, \min_{1\leq i\leq n}(l_i + \max(p_{1,i}, p_{2,i})))$, $j \in J$. As a consequence of Yu's [3] result, $LB^N_1 = C^*_{max}(\tilde{I})$ is a valid lower bound on instance I, which is computed in $O(n \log n)$-time using Mitten [2].

Moreover, we consider two instances $I = (J, p_1, l, p_2)$ and $I' = (J', p_1, l, p_2)$ of $F2|l_j|C_{max}$, where $J' \subset J$. Then, any valid lower bound on I' is also a valid lower bound on I. A new lower bound called LB^N_2 can be obtained by invoking LB^N_1 on different sub-instances of I. Interestingly, we consider n sub-instances. We start by the original instance I, the next sub-instance is built from the one in hand by removing the job that has the minimum value of $l_j + \max(p_{1,j}, p_{2,j})$, $j \in J$.

3 Linear Programming-Based Lower Bound

We consider a mathematical formulation that is based on determining the precedence relationships between jobs on the two machines where it is supposed that the jobs are continuously processed on M_1 and M_2. Indeed, any valid schedule on an $F2|l_j|C_{max}$ instance can be transformed to a schedule with the same makespan value C where jobs are continuously processed on the two machines from time 0 and from time $C - \sum_{j=1}^{n} p_{2,j}$ on M_1 and M_2, respectively.

The decision variables are defined for each pair of jobs i, $j \in J$, where $X^k_{i,j}$ takes the value 1 if i precedes j on M_k and 0 otherwise, $k \in \{1, 2\}$. Furthermore, $C_{k,j}$ represents the completion time of job j on M_k and the total idle time on M_2 is denoted by L. Using these definitions, the model can be formulated as follows:

$$\textbf{min} \quad L \tag{1}$$

$$\text{s.t.} \quad X^k_{i,j} + X^k_{j,i} = 1, \qquad \forall i,\ j \in J\ i \neq j; k \in \{1, 2\} \tag{2}$$

$$X^k_{i,j} \geq X^k_{i,v} + X^k_{v,j} - 1, \qquad \forall i, j, v \in J; k \in \{1, 2\} \tag{3}$$

$$C_{1,i} = \sum_{j=1}^{n} p_{1,j}.X^1_{j,i} + p_{1,i}, \quad \forall i \in J \tag{4}$$

$$C_{2,i} \geq C_{1,i} + l_i + p_{2,i}, \quad \forall i \in J \tag{5}$$

$$C_{2,i} = L + \sum_{j=1}^{n} p_{2,j}.X^2_{j,i} + p_{2,i}, \ \forall i \in J \tag{6}$$

$$L \geq 0,\ C_{k,j} \geq 0,\ X^k_{i,j} \in \{0, 1\} \ \forall i,\ j \in J,\ k \in \{1, 2\} \tag{7}$$

The objective function (1) minimizes the total idle time on M_2. Constraints (2) ensure that for each pair of jobs, one of them has to precede the other on each machine. Constraints (3) guarantee the absence of cyclic precedence relationships between all

jobs. Constraints (4) and (6) take into account the job's precedence and enforce them to be processed continuously without idle on M_1 and M_2, respectively. In addition, Constraints (5) ensure that a job after being processed on M_1 has to wait its time delay to be executed on M_2. The nature of decision variables L, $C_{k,j}$ and $X^k_{i,j}$ is displayed by Constraints (7).

In order to strengthen the LP relaxation of the model, we propose a valid inequality, which is based on the additional waiting time that a job has to fulfill after being available for processing on M_2. We remark that given a sequence of jobs on M_1, solutions in which the jobs are scheduled on M_2 according to their arrival times are dominant. Therefore, if a job j is preceded by a job i on M_1, then a lower bound on the minimum additional waiting time observed by job j or job i is $w_{i,j}$, where (i) $w_{i,j} = \max(0, l_i + p_{2,i} - p_{1,j} - l_j)$, if $l_i \leq p_{1,j} + l_j$ (ii) $w_{i,j} = \max(0, p_{1,j} + l_j + p_{2,j} - l_i)$, if $l_i > p_{1,j} + l_j$.

A lower bound on the total additional waiting time Δ can be obtained by solving the assignment problem where the assignment costs are $\delta_{i,j},\ i,j \in \{1,\ldots,n\}$. In the following, we describe how the assignment costs are computed. Note that the first scheduled job (resp. the last scheduled job) on M_1 is assumed to be preceded (resp. followed) by a dummy job (job 0, resp. job $n+1$). Obviously, since job 0 cannot precede job $n+1$, and a job cannot precede itself, then we set $\delta_{0,n+1} = \infty$ and $\delta_{j,j} = \infty,\ \forall j \in \{1,\ldots,n\}$.

Remark 1 Let us consider an instance I of $F2|l_j|C_{max}$ and LB (resp. UB) a lower bound (resp. an upper bound) on the value of the makespan. If a schedule of makespan LB exists, then the jobs can be continuously processed without any idle time, from time 0 on M_1 and from time $(LB - \sum_{j=1}^{n} p_{2,j})$ on M_2. Then, we obtain the following assignment costs:

- $\forall i \in \{1,\ldots,n\},\ \delta_{0,i} = \max(0, LB - \sum_{j=1}^{n} p_{2,j} - l_i - p_{1,i})$
- $\forall i \in \{1,\ldots,n\}$, if $\sum_{j=1}^{n} p_{1,j} + l_i + p_{2,i} > UB$, i cannot be processed at the last position on M_1, then $\delta_{i,n+1} = \infty$. Otherwise $\delta_{i,n+1} = 0$

In order to set $\delta_{i,j},\ \forall i,j \in \{1,\ldots,n\},\ i \neq j$, we introduce in the following lemma a new dominance rule.

Lemma 1 *Let $I = (J, p_1, l, p_2)$ be an instance of $F2|l_j|C_{max}$ and two jobs $i,j \in J$ such that $p_{1,j} + l_j \leq p_{1,i} + l_i \leq p_{2,j} + l_j$. For any schedule S of I, if j and i are adjacent on M_1 then j should precede i on M_1.*

Proof Let us suppose that j is executed before i on M_1. First, thanks to the relation $p_{1,i} + l_i \leq p_{2,j} + l_j$, i is ready for processing on M_2 while the processing of job j has not yet ended. Then these two jobs are executed continuously without idle on M_2. Second, since $p_{1,j} + l_j \leq p_{1,i} + l_i$, the operations $O_{2,j}$ and $O_{2,i}$ would have started earlier than if i had preceded j on M_1.

Corollary 1 *Let $I = (J, p_1, l, p_2)$ be an instance of $F2|l_j|C_{max}$ and two jobs $i,j \in J$. If $p_{1,j} + l_j \leq p_{1,i} + l_i \leq p_{2,j} + l_j$, then $\delta_{i,j} = \infty$. Otherwise $\delta_{i,j} = w_{i,j}$.*

Similarly, by interchanging the role of M_1 and M_2, we obtain Δ' another lower bound on the total additional waiting time. Therefore, the following valid inequality holds: $\sum_{j=1}^{n} C_{2,j} \geq \sum_{j=1}^{n} C_{1,j} + \sum_{j=1}^{n}(p_{2,j} + l_j) + \max(\Delta, \Delta')$.

4 Computational Results

We present in this section the computational results of the new lower bounds and we compare their performance. We test them on a set of six classes A–F that was proposed by Dell'Amico [1]. Furthermore, preliminary computational results conducted on the literature classes show that previous lower bounds give bad performance when time delays are very large compared to processing times. To that aim, we introduce two new classes of instances where the processing times on M_1 and M_2 and the time delays are randomly generated between $[1 \ldots \alpha]$, $[1 \ldots \beta]$ and $[1 \ldots \gamma]$, respectively, where $\alpha = \beta = 20$ and $\gamma = \frac{n}{2}10$ (resp. $\alpha = \beta = 100$ and $\gamma = \frac{n}{2}100$) for class 1 (resp. class 2). For each class, the number of jobs is $n = 10, 30, 50, 100, 150, 200$. For each combination of class and number of jobs, 10 instances were randomly generated. All algorithms were coded in C++ and compiled under CentOS 6.6. Moreover, we used CPLEX 12.6 to implement the linear programming-based lower bound. The experiments were conducted on an Intel(R) Xeon(R) @ 2.67 GHz processor. For pages limitation, we interest only to the most competitive lower bounds.

In the following, we denote by LB_3^N the optimal objective value that is obtained after solving the LP relaxation of the mathematical model (1)–(7) including the valid inequality and by LB_4^N a version of LB_3^N without Constraints (3). We conducted preliminary computational results on LB_3^N and LB_4^N. Clearly, LB_3^N dominates LB_4^N. However, LB_4^N offers a good trade-off between effectiveness and efficiency. Indeed, for all the considered instances where $n < 100$, LB_4^N achieves the same lower bound values as LB_3^N within a very short time. The average computational time of LB_4^N on these instances is 0.77 s while LB_3^N needs 61.54 s. Furthermore, LB_3^N fails to solve all large scale instances (i.e. $n \geq 100$) within 1800 s, while LB_4^N solves them in an average time of 1.47 s.

In order to present a detailed image of the performance of lower bounds LB_3, LB_4, LB_5, LB_6, LB_2^N and LB_4^N, a pairwise comparison between them is given in Table 1. In this table, we illustrate for each pair of lower bounds LB_{row} and LB_{col}, which are displayed in some given row and column, respectively, the percentage of times where $LB_{col} > LB_{row}$. We observe on classes A–F that LB_2^N outperforms LB_5 in 10.83% of instances and LB_6 in 26.38% of instances. However, on the new classes 1–2, we notice that LB_4^N provides a much better performance than the rest, since it outperforms LB_5 and LB_2^N in 77.5% and 75% of instances, respectively.

To get a better picture of the lower bounds performance, we provide in Table 2 the average percentage deviation (over the instances of each class) with respect to the maximal lower bound value, that is delivered by the considered lower bounds. Note that (-) means that the average CPU time is less than 10^{-2} s. From Table 2, we

Table 1 Pairwise comparison between lower bounds

	Classes A–F						Classes 1–2					
	LB_3	LB_4	LB_5	LB_6	LB_2^N	LB_4^N	LB_3	LB_4	LB_5	LB_6	LB_2^N	LB_4^N
LB_3	–	63.33	99.44	98.05	99.44	100	–	46.66	60.83	52.5	61.66	100
LB_4	36.66	–	99.72	98.33	99.72	100	50	–	61.66	54.16	61.66	100
LB_5	0.55	0.27	–	3.33	10.83	2.5	39.16	38.33	–	7.5	52.5	77.5
LB_6	1.94	1.66	23.33	–	26.38	2.77	47.5	45.83	67.5	–	72.5	79.16
LB_2^N	0.55	0.27	0	0	–	1.94	38.33	38.33	0	0	–	75
LB_4^N	0	0	97.5	97.22	98.05	–	0	0	20.83	19.16	23.33	–

Table 2 Relaxation performance by class

Class	LB_3		LB_4		LB_5		LB_6		LB_2^N		LB_4^N	
	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time	Gap	Time
A	29.9	–	30.26	–	0.04	–	0.3	–	**0**	–	19.81	0.82
B	27.8	–	28.26	–	0.06	–	0.33	–	**0.01**	–	17.82	0.81
C	23.96	–	24.16	–	0.89	–	2.56	–	**0.69**	–	13.73	0.79
D	32.24	–	32.29	–	0.04	–	0.03	–	**0**	–	21.46	0.84
E	53.96	–	46.26	–	0.003	–	0.02	–	**0**	–	33.09	0.82
F	53.32	–	45.85	–	0.02	–	0.22	–	**0**	–	32.93	0.85
1	10.35	–	11	–	1.78	–	2.19	–	1.56	–	**0.92**	0.58
2	10.42	–	10.42	–	13.89	–	19.26	–	13.66	–	**0.11**	0.69
Avg	30.24	–	28.56	–	2.09	–	3.11	–	**1.99**	–	17.48	0.77

observe that the average gaps significantly depend on the classes. On one hand, LB_2^N exhibits an average gap of 1.99% on all classes. However, for the instances of class 2, its average gap jumps to 13.66%. On the other hand, LB_4^N presents a much better performance on the new classes. Indeed, the average gap of this bound is equal to 0.92% and 0.11% on class 1 and class 2, respectively.

5 Conclusion

This paper addressed the two-machine flow shop problem with time delays. We recalled the lower bounds of the literature and proposed new ones. In particular, the linear relaxation of a mathematical formulation with the consideration of a valid inequality and a dominance rule provides the best performance on a set of 120 new instances. Future research needs to be focused on investigating new valid inequalities and dominance rules in order to improve the resolution of the considered model.

Acknowledgements This work is carried out in the framework of the Labex MS2T (Reference ANR-11-IDEX-0004-02). It is also partially supported by the ATHENA project (ANR-13-BS02-0006-02).

References

1. Dell'Amico, M.: Shop problems with two machines and time lags. Oper. Res. **44**(5), 777–787 (1996)
2. Mitten, L.G.: Sequencing n jobs on two machines with arbitrary time lags. Manag. Sci. **5**(3), 293–298 (1959)
3. Yu, W.: The two-machine flow shop problem and the one-machine total tardiness problem. Ph.D. thesis, Eindhoven University of Technology, The Netherlands (1996)

Efficient Ship Crew Scheduling Complying with Resting Hours Regulations

Anisa Rizvanolli and Carl Georg Heise

Abstract To ensure safe and efficient ship operations a proper schedule of crew tasks is necessary. This encompasses a work plan for the crew, consisting of appropriately qualified seafarers, which also complies with the rules of the Maritime Labour Convention (MLC). The optimized crew schedule can reduce crew costs for shipping companies and also help to avoid expensive ship detentions by port state authorities due to incompliances in the crew's work plan. A mathematical model is presented for the crew scheduling problem, which is subject to complex rule sets for working and resting hours. In this model the mandatory tasks for safe ship operation and the crew qualification requirements for these tasks represent the main input parameters. They depend on variables such as the ship type and route and may differ substantially. Furthermore, the model considers common watch-keeping patterns and special constraints on mandatory tasks. This problem is formulated as mixed integer linear program. Numerical experiments with different small real data sets from business practice are also presented.

1 Introduction to the Ship Crew Scheduling Problem with Resting Hours Constraints

Efficient crew scheduling is crucial for safe ship operation during a given voyage. With the main goals being to ensure safety, to avoid accidents, and to establish a reliable management of working and resting hours legislative regulations [5] have to be taken into account during a ship's voyage. For shipping companies crew costs are a considerable part of the total costs. Even more expensive are accidents, damages due to crew fatigue, and fines for being incompliant with the legislative resting regulations. A rested crew that executes all mandatory tasks during a ship's

A. Rizvanolli (✉)
Fraunhofer Center für Maritime Logistik und Dienstleistungen, Hamburg, Germany
e-mail: anisa.rizvanolli@tuhh.de

C.G. Heise
Institut für Mathematik, Technische Universität Hamburg, Hamburg, Germany
e-mail: carl.georg.heise@tuhh.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_71

voyage is therefore critical. The challenge is to find a minimal crew constellation for a container ship and a given voyage, without violating the complex resting hours regulations. At the moment pre-planning and on-board staff scheduling during the voyage are done independently and manually based on the officer's experience.

Similar staff scheduling problems [9] have been treated in many different industry sectors [2] from the airline industry [3], public transportation, call centers [1] to nurse scheduling. They are generally known as cyclic scheduling problems and each of them has quite specific requirements on working shifts and off-days or off-hours. In the shipping industry a similar problem is described in [8]. The main difference between the model presented there and the one treated in this paper lies in the resting hours requirements and the way the qualifications of seafarers are matched with the tasks.

In this paper we present the problem of determining the minimal crew for a container ship and a given voyage, so that all complex resting hours conditions hold for each planned crew member—the *ship crew scheduling problem with resting hours constraints*. A voyage is defined as a sequence of ports together with estimated times of arrival and departure. As described in [7] the start and end times of the mandatory tasks are determined by the voyage. Based on positions and certificates certain seafarers are qualified for a given mandatory task or not. The duration of the tasks can be smaller or equal than the time-span between begin and end and the seafarers planned for one task may be interrupted. We present a mixed integer linear program formulation of this scheduling problem for fixed, flexible, and consecutive tasks. This combinatorial problem [6] consists of two parts: the assignment of tasks to seafarers and the scheduling part with the complex resting hours constraints. These constraints distinguish this problem strongly from the traditional day-off and cyclical scheduling problems, where rest periods are given with fixed start and end times. The length of the planning horizon is determined by the voyage and we quantize the planning horizon into half hour units.

Mandatory tasks and seafarers qualification: mandatory tasks can be assigned to more than one seafarer. Each task can only be assigned to one seafarer simultaneously and only qualified seafarers can be assigned to a given task. Seafarers cannot be scheduled for different tasks in parallel.

Resting hours constraints: Based on the MLC 2016 [5] the following rules must hold for each seafarer and for each half hour time interval during the whole voyage. A seafarer must rest at least 10 h in every 24 h interval. The maximum working hours for a week are 91. The 10 h of rest in 24 h must be at most divided into 2 blocks and one of the blocks must consist of at least 6 h. In the case of a rest being longer than 10 h, a division of it into more than two blocks is allowed, presuming that for 10 h the above constraint holds. The minimal duration of a rest period is 1 h.

2 Mathematical Formulation

The scope of this mixed integer linear program is to determine the minimal crew needed for a container ship for a given port sequence, such that the work schedule of each crew member is compliant with the working/resting hours regulations and has minimal interruptions during the work. In the following the input parameters and variables needed for the MILP formulation are presented.

Parameters

S	The set of all seafarers.
J	The set of all mandatory tasks.
S_j	The set of all seafarers that are qualified for task $j \in J$.
T	The time horizon, i.e. the set of all 30 min long time intervals that are to be planned; $T = \{1, \dots, t_{\max}\}$ for some $t_{\max} \in \mathbb{N}$.
$a_j, e_j \in T$	The start and end time of task $j \in J$, respectively.
$d_j \in T$	The duration of task $j \in J$ in time intervals, satisfying $d_j \leq e_j - a_j + 1$.
$m_{\min} \in T$	The minimum length of a rest period; $m_{\min} = 2$.
$M \subseteq T$	The set of all possible durations for a rest period in 30 min time intervals; $M = \{m_{\min}, \dots, t_{\max}\}$.
$m_C \in T$	The minimum length (in time intervals) for at least one consecutive rest period in each 24 h interval; $m_C = 12$ and $m_C \geq m_{\min}$.
$m_D \in T$	The minimum length (in time intervals) that each seafarer needs to rest in each 24 h interval; $m_D = 20$ and $m_D \geq m_C$.
$m_W \in T$	The minimum length (in time intervals) that each seafarer needs to rest in each one week interval; $m_W = 154$.
$\varepsilon \in \mathbb{R}$	An arbitrary constant that satisfies $0 < \varepsilon < 1$ (e.g., choose $\epsilon = 0.9$).
$T^* \subseteq T$	The set of all time intervals such that the next $49 - m_D = 29$ time intervals (including this interval) are still within the time horizon, implying that minimum rest period lengths need to be checked; $T^* := T \cap [1, t_{\max} - 48 + m_D] = T \cap [1, t_{\max} - 28]$.

Variables

z_s	is 1 if seafarer $s \in S$ is active, otherwise 0.
x_{sjt}	is 1 if seafarer s is executing task j at the t-th time interval, otherwise 0.
w_{sjt}	is 1 if $x_{sjt} = 1$ and $x_{sj(t+1)} = 0$, otherwise 0.
b^m_{st}	is 1 if at the t-th time interval a rest period of length m begins for seafarer s, otherwise 0.
u_{st}	The length of one rest period, capped at m_D, that satisfies the 6-h-condition for seafarer s during the 24 h time interval starting from the i-th time interval. ($s \in S, t \in T^*$).

The following function represents the objective function to be used in the problem setting. Note that we also minimize the number of schedule switches to minimize interruptions during work.

$$\min\left(\sum_{s\in S} z_s\right) + \varepsilon\left(|S|\cdot|T|\right)^{-1}\cdot\left(\sum_{j\in J}\sum_{s\in S_j}\sum_{t=a_j}^{\min(e_j,t_{\max}-1)} w_{sjt}\right)$$

Constraints

$$0 \le z_s \le 1\ \forall s \in S; \qquad 0 \le b_{st}^m \le 1\ \forall s \in S, t \in T \cup \{t_{\max}+1\}, m \in M$$
$$m_C \le u_{st} \le m_D\ \forall s \in S, t \in T^*; \qquad 0 \le x_{sjt} \le 1\ \forall s \in S, j \in J, t \in T$$
$$0 \le w_{sjt} \le 1\ \forall s \in S, j \in J, t \in T \cap [a_j, \min(e_j, t_{\max}-1)] \tag{1}$$

The constraints in Eq. 1 are the basic domains for the variables. *All* variables are integers. Below we present the block of constraints for the assignment of tasks to seafarers without involving any resting hours constraints. Constraint 2 ensures each seafarer only executes tasks if he is active and Constraints 3 and 4 guarantee that each task is executed for its whole length and only by qualified seafarers. With Constraints 5 and 6 no parallel assignment of the same task to different seafarers is allowed and each seafarer can be scheduled for only one task at a time. Constraint 7 ensures that tasks are assigned only between the given begin and end times. Constraint 8 together with the second part in the objective function ensures that tasks are assigned with as few interruptions or (changes of the executing seafarer) as possible.

$$\sum_{j\in J}\sum_{t\in T} x_{sjt} \le t_{\max}\cdot z_s\ \forall s \in S \tag{2}$$

$$\sum_{s\in S_j}\sum_{t=a_j}^{e_j} x_{sjt} = d_j\ \forall j \in J \tag{3}$$

$$\sum_{j\in J}\sum_{s\in S\setminus S_j}\sum_{t\in T} x_{sjt} = 0 \tag{4}$$

$$\sum_{j\in J} x_{sjt} \le 1\ \forall s \in S, t \in T \tag{5}$$

$$\sum_{s\in S} x_{sjt} \le 1\ \forall j \in J, t \in T \cap [a_j, e_j] \tag{6}$$

$$\sum_{j\in J}\sum_{s\in S_j}\sum_{t\in T\setminus[a_j,e_j]} x_{sjt} = 0 \tag{7}$$

$$w_{sjt} \ge x_{sjt} - x_{sj(t+1)}\ \forall j \in J, s \in S_j, t \in T \cap [a_j, \min(e_j, t_{\max}-1)] \tag{8}$$

The following two constraints represent the general resting hours regulations. Constraints 9 and 10 ensure that each seafarer rests at least $m_D/2 = 10$ h each day and at least $m_W/2 = 77$ h each week.

$$\sum_{i=t}^{\min(t_{\max},t+47)} \sum_{j\in J} x_{sji} \leq 48 - m_D \;\forall s \in S, t \in T^* \tag{9}$$

$$\sum_{i=t}^{\min(t_{\max},t+335)} \sum_{j\in J} x_{sji} \leq 336 - m_W \quad \forall s \in S, t \in T \cap [1, t_{\max} - 336 + m_W] \tag{10}$$

Finally, we introduce the constraints for the rest blocks. Constraint 11 ensures that tasks are not assigned during rest periods for each seafarer. Constraint 12 ensures that each rest period has only one beginning and is not counted twice. Constraint 13 connects the u_{st} variables with the b^m_{st} variables, so that $u_{st} \geq m$ only if a sufficient rest period of length m occurs in the corresponding 24 h time interval. Constraint 14 ensures that each seafarer has one or two consecutive rest periods of combined length at least $m_D/2 = 10$ h per 24 h interval.

$$\sum_{i=t}^{\min(t+m-1,t_{\max})} \sum_{j\in J} x_{sji} \leq \min(m, t_{\max} - t + 1) \cdot (1 - b^m_{st}) \;\forall s \in S, t \in T, m \in M \tag{11}$$

$$\left(\sum_{i=1}^{\min(t+m-1,t_{\max}+1)} \sum_{n=\max(t-i+1,m_{\min})}^{t_{\max}} b^n_{si}\right) - b^m_{st} \leq (1 - b^m_{st}) \left\lceil \frac{\max(t+m-1, t_{\max})}{m_{\min}+1} \right\rceil$$
$$\forall s \in S, t \in T \cup \{t_{\max}+1\}, m \in M \tag{12}$$

$$\sum_{i=1}^{\min(t_{\max}+1,t+48-m)} \sum_{n=\max(t-i,0)+m}^{t_{\max}} b^n_{si} \geq \frac{u_{st} - m + 1}{m_D - m + 1}$$
$$\forall s \in S, \forall m \in M \cap [m_C, m_D], t \in T^* \tag{13}$$

$$\sum_{i=1}^{\min(t_{\max}+1,t+48-\hat{m})} \sum_{n=\max(t-i,0)+\hat{m}}^{t_{\max}} b^n_{si} \geq \frac{m - u_{st} + 1}{m - m_C + 1} + 1$$
$$\forall s \in S, \forall m \in M \cap [m_C, m_D - 1], t \in T^*, \hat{m} := \max(m_{min}, m_D - m) \tag{14}$$

The model can be further enhanced with some additional cuts to speed up the optimization, though due space constraints we do not present them here.

3 Numerical Results and Conclusion

In this section we present a small instance with 12 tasks to be scheduled among 4 deck officers. These tasks include watch-keeping given in a 4/8 pattern and further mandatory tasks needed to be scheduled during approaching, staying, and leaving

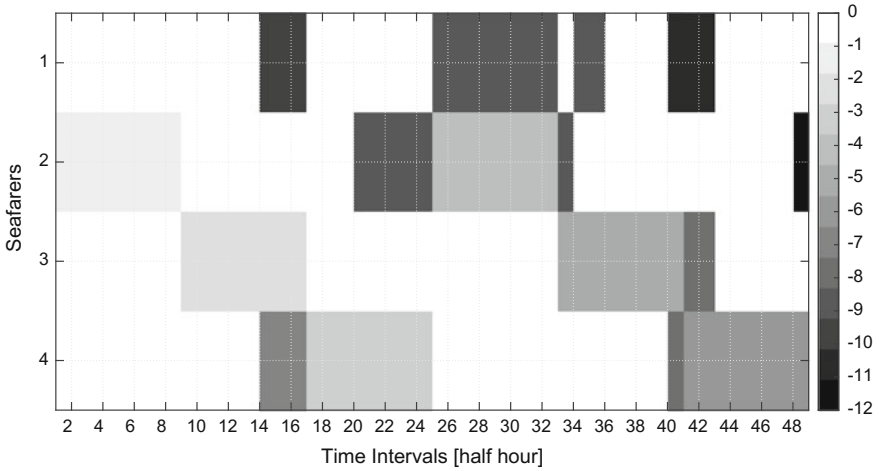


Fig. 1 Work schedule for 4 Deck Officers with 12 mandatory tasks during a port approach, stay and leave. The white spaces indicate rest times

a port. This instance as well as all other schedules are calculated using IBM ILOG CPLEX Optimization Studio v12.6 [4]. The computation time for this instance is 0.47 s. For the same instance size but without watch-keeping pattern the computation time was 9 min. Extending the planning horizon by 12 h, we realize that it is not possible to cover the watch-keeping pattern requirements and the total work load with the given manpower. We get a feasible work schedule with 4 Officers only if we resign the pattern. In that case the computational time increases to 5.6 h. With the increase of number of tasks and size of the sets with qualified seafarers for each task, the computation time increases very fast (Fig. 1).

In this paper we presented the MILP formulation for the crew scheduling problem with complex resting hours constraints. With additional cuts and a feasible good start solution quite big instances of the problem can be solved or at least further improved in realistic time frames. First experiments have shown that the additional cuts save half of the computation time needed to solve the model without them, at the cost of increasing the total number of constraints. Calculations with small real world data instances can still be a very good base for decision makers and a first step toward optimization of crew scheduling in the shipping industry. A lot of knowledge and information can be gained from these results regarding optimal watch-keeping patterns for officers. Furthermore, this model enables for the first time the automatic check of the complex working resting hours for each seafarer planned on the ship.

References

1. Çezik, T., Günlük, O., Luss, H.: An integer programming model for the weekly tour scheduling problem. Naval Res. Logist. **48** (2001)
2. Ernst, A.T., Jiang, H., Krishnamoorthy, M., Owens, B., Sier, D.: An annotated bibliography of personnel scheduling and rostering. Ann. Oper. Res. **V1–4**, 21–144 (2004)
3. Gopalakrishnan, B., Johnson, E.L.: An annotated airline crew scheduling: state-of-the-art. Ann. Oper. Res. **1**, 305–337 (2005)
4. IBM Corporation: IBM ILOG CPLEX Optimization Studio V12.6.1 documentation (2014)
5. International Maritime Organisation: Maritime Labour Convention Regulation 2.3 (2016). http://www.ilo.org/wcmsp5/groups/public/@ed_norm/@normes/documents/normativeinstrument/wcms_090250.pdf
6. Korte, B., Vygen, J.: Combinatorial optimization—Theory and Algorithms, 4th ed. Springer (2008)
7. John, O., Gailus, S., Rizvanolli, A., Rauer, R.: An integrated decision support tool for hours of work and rest compliance optimization during ship operations. In: Bertram, V. (eds.) 13th International Conference on Computer and IT Applications in the Maritime Industries. Compit 1. Band. Hamburg (Schriftenreihe Schiffbau), pp. 245–256 (2014)
8. Li, H.: A Decomposition approach for shipboard manpower scheduling. Military Oper. Res. Mil. Oper. Res. **14** (2009)
9. Pinedo, M.L.: Scheduling—Theory, Algorithms, and Systems, 4th ed. Springer (2012)

A Multi-criteria MILP Formulation for Energy Aware Hybrid Flow Shop Scheduling

Sven Schulz

Abstract Managing energy consumption more sustainably and efficiently has been gaining increasing importance in all industrial planning processes. Energy aware scheduling (EAS) can be seen as a part of that trend. Overall, EAS can be subdivided into three main approaches. In detail, the energy consumption can be reduced by specific planning, time-dependent electricity cost might be exploited or the peak power may be decreased. In contrast to the majority of EAS models these ideas are adopted simultaneously in the proposed new extensive MILP formulation. In order to affect peak load and energy consumption, variable discrete production rates as well as heterogeneous parallel machines with different levels of efficiency are considered. As a result, the interdependencies of different energy aware scheduling approaches and especially a dilemma between peak power minimization and demand charge reduction can be shown.

1 Introduction

To reduce electricity demand, companies normally invest in new technologies and processes. However, with intelligent scheduling we are also able to reduce energy demand and costs without losing productivity. Moreover, scheduling has two major advantages: firstly, no high investments are necessary and secondly, it can be implemented immediately.

There are three different strategies in energy-aware scheduling (EAS) which can be pursued to reduce energy costs. *Reducing energy consumption* directly is the first approach. Such savings can be achieved by selecting parallel machines with low energy consumption, by decreasing production speed or by taking advantage of different machine states like idle or standby (intelligent on/off decisions). A second

S. Schulz (✉)
TU Dresden, 01062 Dresden, Germany
e-mail: sven.schulz@tu-dresden.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_72

strategy is to *make use of time depending energy prices*. By shifting energy consumption from peak price times to times of lower energy prices, energy costs can be reduced while energy consumption stays at the same level. Besides the consumption charge, companies often pay also a demand charge for the maximum power demand during the billing period. A third approach in EAS is now to level the energy needs in order to *lower the peak power* and hence the demand charge.

Only a handful hybrid flow shop problems consider some of the mentioned approaches. In [2] an EAS problem can be found, whereby the peak power is gradually reduced on the basis of an APS-system. Whereas [6] publish a hybrid flow shop problem with variable machine speed and time depending electricity prices, an approach for on-off decisions for a closed loop flow shop plant is presented in [7]. Also [3] consider different machine states in a flexible flow shop problem to reduce energy consumption and makespan simultaneously. Tan et al. [9] describe a two-stage mathematical programming approach to solve a parallel hybrid scheduling problem in steel making process with variable energy prices.

To the best of our knowledge, there is no paper considering the three mentioned basic strategies simultaneously. The primary concern of this paper is to analyze the effects and interdependencies of different EAS measures. Therefore, a comprehensive MIP including a wide range of energy-aware aspects is developed in Sect. 2. In Sect. 3 a numerical example serves to illustrate how the model operates as well as to visualize the interdependencies in EAS. Section 4 gives a short summary.

2 A Comprehensive MIP for EAS

Indices

j	Job in J
m	Machine in M_s
s	Production stage in S
t	Time period in T
v	Speed level in V

Parameters

P_{max}	Maximum peak power
E_{sj}^{m}	Energy consumption
D_j	Due date
R_j	Release date
S_{sj}^{m}	Standard processing time
ce_t	Electricity cost
cp_j	Production cost
ct_j	Tardiness cost
a_s^{mv}	Energy savings

Decision Variables

$c_{sj} \in \mathbb{N}$	Completion time of task s of job j
$g_{sj}^{mv} \in \{0,1\}$	Processing time extension v of task s of job j on machine m
$T_j \in \mathbb{N}$	Tardiness of job j
$p_{sjt}^{m} \in \mathbb{N}$	Power consumption of task s of job j on machine m in time period t
$x_{sjt}^{m} \in \{0,1\}$	Task s of job j is performed on machine m in time period t
$y_{sj}^{m} \in \{0,1\}$	Task s of job j is assigned to machine m
$z_{sjt}^{m} \in \{0,1\}$	Execution of task s of job j on machine m starts in time period t

Every job j has to be processed at each production stage and in every stage s there is a set of unrelated parallel machines denoted as M_s. The planning horizon is divided into T_{max} time-intervals of equal length. Using the introduced notation above the EAS Mixed Integer Problem can be modelled as follows:

Minimize
$$\sum_{j\in J}(ct_j \cdot T_j + cp_j \cdot (c_{S_{max}j} - (R_j - 1))) + \sum_{t\in T}(ce_t \cdot \sum_{j\in J}\sum_{s\in S}\sum_{m\in M_s} p_{sjt}^{m}) \quad (1)$$

Subject to:
$$\sum_{j\in J} x_{sjt}^{m} \leq 1 \quad \forall\, s,m,t \quad (2)$$

$$\sum_{t\in T}\sum_{m\in M_s} z_{sjt}^{m} = 1 \quad \forall\, j,s \quad (3)$$

$$x_{sjt}^{m} \leq z_{sjt}^{m} \quad \forall\, j,s,m,t = R_j \quad (4)$$

$$x_{sjt}^{m} - x_{sj,t-1}^{m} \leq z_{sjt}^{m} \quad \forall\, j,s,m,t \geq R_j \quad (5)$$

$$\sum_{t\in T} x_{sjt}^{m} = S_{sj}^{m} \cdot y_{sj}^{m} + \sum_{v\in V} g_{sj}^{mv} \quad \forall\, j,s,m \quad (6)$$

$$\sum_{m\in M_s} y_{sj}^{m} = 1 \quad \forall\, j,s \quad (7)$$

$$\sum_{t\in T|t\geq R_j}\sum_{m\in M_s}(z_{sjt}^{m} - z_{s-1,jt}^{m}) \cdot t \geq \sum_{m\in M_{s-1}}(S_{s-1,j}^{m} \cdot y_{s-1,j}^{m} + \sum_{v\in V} g_{sj}^{mv}) \quad \forall\, j,s>1 \quad (8)$$

$$\sum_{t\in T|t\geq R_j}\sum_{m\in M_s} z_{sjt}^{m} \cdot t = c_{sj} - \sum_{m\in M_s}(S_{sj}^{m} \cdot y_{sj}^{m} + \sum_{v\in V} g_{sj}^{mv}) + 1 \quad \forall\, j,s \quad (9)$$

$$T_j \geq c_{sj} - D_j \quad \forall\, j,s \quad (10)$$

$$p_{sjt}^{m} = \max(0; E_{sj}^{m} \cdot (x_{sjt}^{m} - \sum_{v\in V} g_{sj}^{mv} \cdot a_{sj}^{mv})) \quad \forall\, j,s,m,t \quad (11)$$

$$\sum_{v \in V} g_{sj}^{mv} \leq S_{sj}^{m} \cdot y_{sj}^{m} \quad \forall j, s, m \tag{12}$$

$$g_{sj}^{m,v-1} \geq g_{sj}^{mv} \quad \forall j, s, m, v > 1 \tag{13}$$

$$\sum_{j \in J} \sum_{s \in S} \sum_{m \in M_s} p_{sjt}^{m} \leq P_{max} \quad \forall t \tag{14}$$

(2) ensures that every machine can process only one job in each period. Since non-preemption is assumed, (3) guarantees that each job has only one starting time period at each production stage. (4) and (5) are necessary to determine z_{sjt}^{m} depending on x_{sjt}^{m}. We introduce (6) to accurately reflect machining time that consists of standard processing time and extra time caused by production speed reductions (g_{sj}^{mv}). Hereby, y_{sj}^{m} serves to select machine m for each job at production stage s. With (7) every job is exactly allocated to one machine at each stage. As a matter of course, no job can be scheduled on a machine before the previous job on this machine is completed (8). (9) serves to calculate the completion time of a task, the tardiness of a job finally results from (10).

Since it is assumed that the parallel machines have different energy efficiencies for different jobs, energy consumption can be reduced by assigning jobs to machines with lower demand. Energy demand may also be reduced by decreasing production speed. Therefore, in (6) and (8) the possibility of increasing the manufacturing time gradually is already considered by g_{sj}^{mv}. Depending on the additional time energy consumption is reduced by the percentage a_{sj}^{mv} in (11).

While energy conversion efficiency is very high for power usage greater than 75% of rated load, electric motors operating slower than 50% of maximal speed show excessive wear and energy consumption in relation to the production output. Kaya et al. [5] condition (12) ensures that the cumulative number of speed reductions can never exceed the standard processing time and a throttling higher than 50% is thus avoided. Furthermore, (13) is deployed in order to enable stepwise speed changes while making sure that no speed level is skipped.

An important characteristic of the model is to take advantage of energy price fluctuations. This unavoidably requires a time-dependent electricity price ce_t. Besides the electricity costs, the objective function (1) minimizes delays and total completion time which are multiplied by a cost factor. Since energy costs consist of consumption and demand charges also costs for energy peak should be considered.

Peak load charges have to be paid for long time periods (quarterly, yearly). In contrast, scheduling is used most commonly for the purpose of operational decision-making and it normally examines shorter periods (daily, weekly). Therefore, optimizing peak load costs within the scheduling model is only advisable if the billing period corresponds to the period considered. The approaches of [1] or [4] are examples of models that directly include energy peak costs in the objective function.

In this contribution a different approach is pursued. Due to the usually unequal time periods of peak load charges and scheduling horizon it is preferred to integrate

the peak load as a constraint into our model. Often the maximum peak power is known from the past. Constraint (14) ensures that energy consumption is always lower than this value. By varying P_{max} a Pareto front can be developed and the peak load can be improved too.

3 Computational Experiments

A two-stage hybrid flow shop with two parallel machines on each stage shall serve as an example. Eight jobs with non-identical release and due dates are considered. All examples are based on randomly generated parameters within given ranges.

In order to give a high incentive to meet due dates, the tardiness cost parameter ct_j is put at 500 for each job j. Obviously, the ratio between production and energy costs is of substantial importance. It is assumed that 50% of the variable costs are energy costs. To allow the energy costs to be as realistic as possible, Phelix spot market prices (15 August 2015) are used. The prices are depicted in Fig. 1. Considering the average energy consumption, the production cost factor is assumed to be 100. Additionally, the energy savings depending on production speed reductions are required. For the example electric motor energy savings following [8] are discretized. The EAS-model is solved using IBM ILOG CPLEX. All problem instances are tested on an Intel Xeon, 3.46 GHz computer.

Leaving aside (14) leads to an energy peak of 45. Based on this value, P_{max} will be parametrically reduced. To keep the calculation time low, the optimal costs of the previous instance are always the lower bound for the next lower peak power scenario. Selected parts of the results are represented in Table 1.

At first, P_{max} can be reduced without any changes in the results. After this initial reduction total costs increase with lower peak power. Therefore, it must be kept in mind that peak power charges decline with lower peak power and these charges are not included in the total costs. Moreover, a diminishing maximum peak power goes along with decreasing energy demand, while makespan, delays and computing time increase. The energy costs tend to decrease with shrinking maximum peak power.

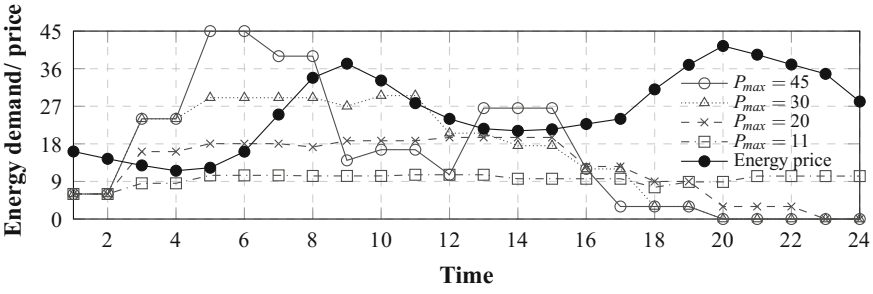


Fig. 1 Energy consumption for different peak power scenarios and real time energy price (RTP)

Table 1 Selected solutions of the numeric example

P_{max}	45	39	35	30	25	20	15	11
Total cost	17584.8	17776.1	18265.9	18358.3	20282.2	22040.2	25095	32031.4
Energy cost	8384.8	8576.1	9165.9	8558.3	8282.2	7240.2	7195	6031.4
Delays	3	3	3	4	7	11	16	29
E. demand	386.5	370.6	395.5	368.5	344.7	302.2	280.0	228.4
Makespan	19	19	19	19	21	22	21	24
Throttlings	5	5	6	5	10	15	17	24

Nevertheless, due to the volatile energy prices it is possible that lower energy demand leads to higher energy costs.

Peak power reduction causes postponements and more production speed throttling and hence lower total energy demand. The load curves in Fig. 1 illustrate the effects on energy demand. By taking a closer look at the curves it can be noted, that the possibilities of taking advantage of energy price fluctuations decrease with lower peak power. This is due to the leveling effect on the energy consumption that goes along with lower P_{max}-values. The example illustrates what theoretically has already been explained. Reducing energy peak and making use of time depending energy prices are contrary objectives (energy cost dilemma).

Besides this insight also the general influence of energy cost consideration and variable production speed shall be examined. Therefore, our EAS-model will be solved further three times disregarding certain aspects. All scenarios are put into relation with the basic model as regards our cost-oriented objective function and the energy consumption. The results are shown in Fig. 2.

As might be expected, ignoring some aspects leads to higher total costs in all scenarios. The influence of time-depending energy prices is relatively low. For $P_{max} = 45$ the costs are 1.6% higher, but for lower peak power the gap is less than 1%. Interestingly, the scenario with constant energy prices leads to less energy consumption.

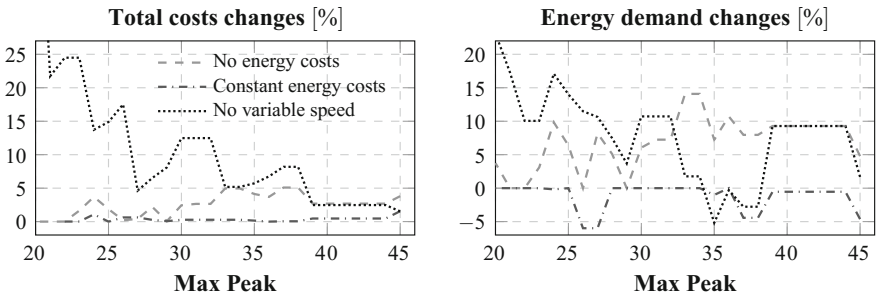


Fig. 2 Cost and energy demand changes with problem variations

Without any energy price considerations costs increase up to 5% and energy consumption up to 14%. This means that the consideration of energy costs can lead to significant savings. Large percentage deviations can occur, if the production speed is assumed to be fixed. Facing low peak power limits, speed changes are especially important to avoid cost increases up to 25%.

4 Conclusions and Further Research

In this paper a cost-oriented energy-aware MILP-model was developed to solve hybrid flow-shop problems. Recently, several articles have been published concerning EAS. However, to the best of our knowledge none of them simultaneously examines all three basic strategies of EAS mentioned above. To close this research gap, a comprehensive MILP for hybrid flow shop scheduling is formulated. The specific functioning of the model was illustrated by a numerical example and the impact of different EAS strategies was investigated. It could be shown that there are contrary effects in the different approaches of EAS. Especially peak power reduction and exploiting time depending energy prices are contrary objectives.

References

1. Babu, C., Ashok, S.: Peak load management in electrolytic process industries. IEEE Trans. Power Syst. **23**(2), 399–405 (2008)
2. Bruzzone, A., Anghinolfi, D., Paolucci, M., Tonelli, F.: Energy-aware scheduling for improving manufacturing process sustainability: a mathematical model for flexible flow shops. CIRP Ann. Manuf. Technol. **61**(1), 459–462 (2012)
3. Dai, M., Tang, D., Giret, A., Salido, M., Li, W.: Energy-efficient scheduling for a flexible flow shop using an improved genetic-simulated annealing algorithm. Robot. Comput. Integr. Manuf. **29**(5), 418–429 (2013)
4. Fang, K., Uhan, N., Zhao, F., Sutherland, J.W.: A new approach to scheduling in manufacturing for power consumption and carbon footprint reduction. J. Manuf. Syst. **30**(4), 234–240 (2011)
5. Kaya, D., Yagmur, E., Yigit, K., Kilic, F., Eren, A., Celik, C.: Energy efficiency in pumps. Energy Convers. Manag. **49**(6), 1662–1673 (2008)
6. Luo, H., Du, B., Huang, G., Chen, H., Li, X.: Hybrid flow shop scheduling considering machine electricity consumption cost. Int. J. Prod. Econ. **146**(2), 423–439 (2013)
7. Mashaei, M., Lennartson, B.: Energy reduction in a pallet-constrained flow shop through on-off control of idle machines. IEEE Trans. Autom. Sci. Eng. **10**, 45–56 (2013)
8. Saidur, R., Rahim, N., Ping, H., Jahirul, M., Mekhilef, S., Masjuki, H.: Energy and emission analysis for industrial motors in malaysia. Energy Policy **37**(9), 3650–3658 (2009)
9. Tan, Y., Huang, Y., Liu, S.: Two-stage mathematical programming approach for steelmaking process scheduling under variable electricity price. Int. J. Iron Steel Res. **20**(7), 1–8 (2013)

Providing Lower Bounds for the Multi-Mode Resource-Constrained Project Scheduling Problem

Christian Stürck and Patrick Gerhards

Abstract We present lower bounds (LB) for the multi-mode resource-constrained project scheduling problem (MRCPSP). Traditionally, the LB for the MRCPSP are derived from the critical path method (CPM). Here, the mode with the shortest duration of each activity is chosen. We improve these LB. New earliest starting times (EST) are calculated by solving several integer programs with a standard solver. These new EST partially improve the EST calculated by the critical path method. This also reduces the number of variables in the model and, in the best case, proves optimality of the best known solutions. Computational results show that these new starting times provide a tighter bound than the LB obtained from CPM.

1 Introduction

The multi-mode resource-constrained project scheduling problem (MRCPSP) minimizes the makespan of a project. A project consists of several activities which have precedence relations between themselves. They can be executed in different modes. Each mode has a duration and a consumption of a given number of renewable (e.g. manpower) and non-renewable resources (e.g. budget). Each activity has to be assigned to a mode and a starting time, such that all precedence and resource constraints are satisfied.

Lower bounds for the MRCPSP are traditionally computed with the critical path method using the mode with the shortest duration for each activity [9]. While there are several ways for calculating lower bounds for the single mode problem (see [5]), there are only few methods for obtaining tight lower bounds for the multi-mode extension. For the MRCPSP extension with minimum and maximum time lags (MRCPSP/max) a destructive lower bound is presented by Brucker and Knust [1]. For the MRCPSP/max Heilmann [3, 4] presented lower bounds using the properties

C. Stürck (✉) · P. Gerhards
Helmut Schmidt University, Hamburg, Germany
e-mail: christian.stuerck@hsu-hh.de

P. Gerhards
e-mail: patrick.gerhards@hsu-hh.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_73

of the modes of the activities. LP-based lower bounds for the MRCPSP were presented by Maniezzo and Mingozzi [7]. Zhu et al. [13] computed a distance matrix with the distances between each activity in the preprocessing step of a branch and cut algorithm for the MRCPSP. New earliest finishing times were provided with the distance matrix, which led to new lower bounds. Muller [8] used a modified distance matrix as well as adaptations of lower bound techniques for the single mode RCPSP to compute new LB for the MRCPSP.

The goal of our approach is providing new lower bounds for the MRCPSP. We apply these to instances of the MMLIB [12]. To our knowledge, no one else has worked on calculating lower bounds for the MMLIB yet. The procedure of Zhu et al. [13] inspired our approach. They applied their procedure to the PSPLIB [6]. Because the MMLIB instances have more activities with more modes than the PSPLIB, we did not calculate a distance matrix (cf. above). We used a MIP-solver instead of a genetic algorithm to calculate the distance between each activity and their predecessors. With these distances new earliest starting times (EST_{MIP}) of all activities are provided. These adapted starting times lead to a new lower bound for the makespan. Besides providing a new lower bound, the new starting times reduce the number of variables in time indexed models for the MRCPSP.

In Sect. 2 the solution approach is described. A mathematical model is given and the procedure is explained by a small example. In Sect. 3 the computational results are presented. The paper closes with a brief conclusion and an outlook on further research.

2 Solution Approach

The MRCPSP contains a set of non-preemptable activities $A = \{0, \ldots, n+1\}$ with a set $\mathscr{R}$ of renewable and a set $\mathscr{R}^n$ of non-renewable resources. For each activity i there is a set of modes M_i. Depending on the chosen mode m, activity i has a duration $d_{i,m}$ and renewable and non-renewable resource consumption $r^r_{i,m,l}$ and $r^n_{i,m,k}$. The objective is to find a resource and precedence feasible schedule that minimizes the makespan of the project.

In this work, our goal is to determine new earliest starting times (EST) for the activities of the MMLIB instances. We obtain a new lower bound with these new earliest starting times. A solution of our approach is represented by a vector EST_{MIP} of earliest starting times for all activities of the project. Traditionally, the EST of each activity is calculated with the critical path method (CPM), which ignores the resource constraints. We initialize the value of EST_{MIP} for each activity by using CPM. We then try to improve these by using integer programs (IP) which ensure that none of the resource constraints is exceeded by the parallel execution of the predecessors. For each activity with more than one predecessor, an IP given by (1)–(7) is used to determine a feasible schedule with minimal starting times. The current activity is called z. The objective of the IP is to minimize the starting time of z. Thus for all its predecessors, a feasible schedule (with respect to precedence and resource

constraints) has to be determined for the problem (1)–(7). Therefore all predecessors P_z of z are passed to the IP which is stated as follows:

$$\min \sum_{t=EST^z}^{LST^z} z_t \cdot t \tag{1}$$

$$s.t. \quad \sum_{m \in M_i} \sum_{t=EST^i}^{LST^i} x_{i,m,t} = 1 \qquad \forall i \in P_z \tag{2}$$

$$\sum_{m \in M_i} \sum_{t=EST^i}^{LST^i} x_{i,m,t} \cdot (t + d_{i,m}) \leq \sum_{t=EST^z}^{LST^z} z_t \cdot t \qquad \forall i \in P_z \tag{3}$$

$$\sum_{i \in P_z} \sum_{m \in M_i} \sum_{t=EST^i}^{LST^i} x_{i,m,t} \cdot r^n_{i,m,k} \leq a^n_k - MC^z_k \qquad \forall k \in \mathscr{R}^n \tag{4}$$

$$\sum_{i \in P_z} \sum_{m \in M_i} \sum_{q=\max(EST^i, t-d_{i,m}+1)}^{\min(t, LST^i)} x_{i,m,q} \cdot r^r_{i,m,l} \leq a^r_l \quad \forall l \in \mathscr{R}, t = 0, \ldots, LST^z \tag{5}$$

$$z_t \in \{0,1\} \quad \forall t = EST^z, \ldots, LST^z \tag{6}$$

$$x_{i,m,t} \in \{0,1\} \quad \forall i \in P_z, \forall m \in M_i, t = EST^i, \ldots, LST^i \tag{7}$$

The objective function (1) minimizes the starting time of the regarded activity. Therefore, z_t is 1 for the earliest starting time t of activity z. Constraint (2) ensures that all predecessors are assigned to exactly one mode. The next constraint (3) replaces the precedence constraints from the original model [11]. Activity z can only start if all predecessors have finished. In term (4) we subtract MC^z_k from the non-renewable resource availability a^n_k. The term MC^z_k is stated as follows:

$$MC^z_k = \sum_{i \in A \setminus P_z} min\{r^n_{i,m,k} : m \in M_i\} \quad \forall k \in \mathscr{R}^n \tag{8}$$

It is the sum of the minimal consumption of the non-renewable resource k for all activities A of the project, excluding the set of predecessors P_z of z. This subtraction is possible because we know that each activity has to be assigned to exactly one mode during the project. The overall resource consumption must not exceed the remaining resource availability.

The constraint (5) ensures that the capacities of the renewable resources a^r_l are not exceeded by the resource consumptions of the predecessors. In (6) and (7) the binary decision variables are defined. The EST and LST (latest starting times) that are used in the IP are calculated by the critical path method. For the calculation of LST the best known solutions of the instances were used (see Sect. 3).

The value of EST^z_{MIP} is updated with the objective value of the IP. The integer programs of the following activities are then solved with updated starting times.

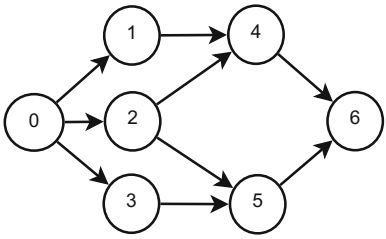


Fig. 1 Network

Table 1 Project information

i	m_i	d_{im}	r^r_{im}	r^n_{im}
0	1	0	0	0
1	1	1	3	2
	2	2	2	3
2	1	1	3	4
	2	4	1	3
3	1	2	2	9
	2	3	3	4
4	1	1	1	6
	2	2	2	4
5	1	1	4	3
	2	2	3	2
6	1	0	0	0

To illustrate our procedure, we consider an example project with 5 non-dummy activities. Each activity has two modes. We have one renewable and one non-renewable resource. The availability of the renewable resource is 5, the availability of the non-renewable resource is 17. The precedence constraints are shown in the network in Fig. 1. In Table 1, the duration $d_{i,m}$ and resource requirements ($r^r_{i,m}$ and $r^n_{i,m}$) of each activity and mode are shown.

The first activity with more than one predecessor is activity 4. First, we have to calculate the minimal resource consumption MC^4_1. All non-dummy variables that are not predecessors of 4 are: 3, 4 and 5. Note that also the minimal consumption of $r^n_{i,m,k}$ for the current activity z is taken into account. The minimal resource consumptions r^n_{im} are 4, 4, and 2 (for the activities 3, 4 and 5). Therefore, MC^4_1 is 10. All predecessors (activity 1 and 2) and MC^4_1 are passed to the MIP-solver. Because of the conflict of the renewable resources r^r_{im}, the $EST^4_{CPM} = 1$ cannot be realised. With activity 1 set to mode 2 and activity 2 to mode 1, the product $z_t \cdot t$ is minimised. The new earliest starting time (EST_{MIP}) of 4 is $EST^4_{MIP} = 2$.

Next, we look at activity 5. We calculate MC_1^5: $2 + 4 + 4 = 8$. The predecessors are activity 2 and 3. Mode 1 of activity 3, which was originally selected by the CPM, cannot be taken in combination with any mode of activity 2. This would exceed the non-renewable resource constraint (available: $17 - 8 = 9$). This leads to an $EST_{MIP}^5 = 4$.

The last activity that we consider is the dummy activity 6, with $MC_1^6 = 9$. With activity 4 executed in mode 2 and activity 5 in mode 1, the EST_{MIP}^6 is minimal with 5. Because of the different (updated) starting times ($EST_{MIP}^4 = 2$ and $EST_{MIP}^5 = 4$), no conflict of the renewable resources occurs. Activity 6 is the end of the project. Thus, $EST_{MIP}^6 = 5$ is the new lower bound of the makespan. This bound is tighter compared to the lower bound obtained by the critical path method ($EST_{CPM}^6 = 3$).

3 Computational Results

We applied this approach to the MMLIB instances [12]. For our experiments we took the best known solutions from the work of Van Peteghem and Vanhoucke [12], Geiger [2] and Stürck et al. [10]. We used a PC with an Intel Xenon CPU at 3.33 GHz and 12 GB of RAM. The implementation of the algorithm was done with C# and CPLEX 12.6.3 as solver for the IP sub-problems.

The computation times were highly depending on the number of predecessors and modes. For 13/12/10 predecessors (with 3/6/9 modes) or less, the IPs were solved within milliseconds. Because there were activities with more predecessors, the MIP-solver was given a time cap of 60 s. If the time cap was reached, the lower bound of the IP found so far was taken. This value was rounded up to the next integer. Because of the design of the MMLIB we know that only integer starting times are valid. The results are shown in Table 2.

For some instances the best known solutions are already equal to the critical path lower bound and thus optimal. These instances were not investigated. This leaves us with 3,477 instances which could be improved. Our procedure was able to update the

Table 2 Computational results

	MMLIB50	MMLIB100	MMLIB+
Number of instances	540	540	3,240
Known optima from the results of [2, 10, 12]	218	248	377
Instances without known optimum	322	292	2,863
Number of instances with improved *EST*	83	50	713
New known optima from EST_{MIP}	2	3	0
New known optima from LP-relaxation	12	9	6
New known optima from LP-relaxation + EST_{MIP}	13	11	6

starting times of 25.78%, 17.12%, 24.90% of the MMLIB50, MMLIB100, MMLIB+ instances, respectively. The algorithm was also able to close the gap to optimality for 2 instances of the MMLIB50 and 3 instances of the MMLIB100. We compared these results with the LP-relaxation of the MIP formulation [12]. Next, we solved the LP-relaxation again but now with the updated starting times EST_{MIP}. While the LP-relaxation could close the gap for more instances than the EST_{MIP} alone, the integration of EST_{MIP} in the LP-relaxation performs best. It was able to close the gap to the best known solutions for 13 instances of the MMLIB50, 11 of the MMLIB100 and 6 of the MMLIB+.

4 Conclusion

We presented a new procedure to compute lower bounds for the MRCPSP. It uses improved earliest starting times of activities derived from solutions of specific IP sub-problems. The computational experiments have shown that new earliest starting times could be calculated for 24.33% of the instances. Much tighter lower bounds (compared to the critical path method) were provided for all problem classes of the MMLIB. The gap to optimality was closed for 27 instances.

Besides of providing lower bounds, the calculation of EST_{MIP} can be used as a preprocessing procedure. Especially exact methods and matheuristics can benefit from the new earliest starting times because they reduce the number of variables in the model (if the model is time indexed). For future work the impact of the EST_{MIP} as a preprocessing element of a matheuristic will be investigated.

References

1. Brucker, P., Knust, S.: Lower bounds for resource-constrained project scheduling problems. Eur. J. Oper. Res. **149**(2), 302–313 (2003)
2. Geiger, M.J.: A multi-threaded local search algorithm and computer implementation for the multi-mode, resource-constrained multi-project scheduling problem. Eur. J. Oper. Res. **256**(3), 729–741 (2017)
3. Heilmann, R.: Ressourcenbeschränkte Projektplanung im Mehr-Modus-Fall. Deutscher Universitäts-Verlag (2000)
4. Heilmann, R.: A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags. Eur. J. Oper. Res. **144**(2), 348–365 (2003)
5. Klein, R., Scholl, A.: Computing lower bounds by destructive improvement: an application to resource-constrained project scheduling. Eur. J. Oper. Res. **112**(2), 322–346 (1999)
6. Kolisch, R., Sprecher, A.: PSPLIB—a project scheduling problem library: OR software—ORSEP operations research software exchange program. Eur. J. Oper. Res. **96**(1), 205–216 (1997)
7. Maniezzo, V., Mingozzi, A.: A heuristic procedure for the multi-mode project scheduling problem based on Benders decomposition. In: Weglarz, J. (ed.) Project Scheduling: Recent Models, Algorithms and Applications, pp. 179–196. Springer (1999)

8. Muller, L.F.: An adaptive large neighborhood search algorithm for the multi-mode RCPSP. Technical report, Report 3.2011, Department of Management Engineering, Technical University of Denmark (2011)
9. Sprecher, A., Hartmann, S., Drexl, A.: An exact algorithm for project scheduling with multiple modes. Oper. Res. Spectr. **19**(3), 195–203 (1997)
10. Stürck, C., Gerhards, P., Fink, A.: A MIP-based adaptive large neighborhood search for the multi-mode resource-constrained project scheduling problem. In: Ruiz, R., Alvarez-Valdes, R. (eds.) Proceedings of the 15th International Conference on Project Management and Scheduling, pp. 243–246. ADEIT Fundaci Universitat Empresa (2016)
11. Talbot, F.B.: Resource-constrained project scheduling with time-resource tradeoffs: the non-preemptive case. Manag. Sci. **28**(10), 1197–1210 (1982)
12. Van Peteghem, V., Vanhoucke, M.: An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. Eur. J. Oper. Res. **235**(1), 62–72 (2014)
13. Zhu, G., Bard, J.F., Yu, G.: A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. INFORMS J. Comput. **18**(3), 377–390 (2006)

Part XVII
Security and Disaster Management

A Macroscopic System Dynamics Model for a Generic Airport

G. Barbeito, M. Moll, S. Pickl and M. Zsifkovits

Abstract The overall dynamics of an airport are multifaceted and very complex. With the ever increasing number of visitors everyday it is important to understand their behaviour. In this paper we present a new macroscopic system dynamics model of the overall workings of a generic airport. The model follows passengers through to the gates modeling various different behaviors on the way there. It also includes implicitly the fleet composition, the number of lanes and explicitly the impact on the noise level. Extra effort was taken to allow for the inherent stochasticity of many of these multi-layered processes. To make it more flexible on- and off-peak times are implemented as well. Moreover random extreme events in the form of emergency landings and heightened security levels have been included. First results provide insight to the change in system behavior under these circumstances.

1 Introduction

With the number of air travel passengers ever increasing—from 3.2 Billion in 2014 to 3.4 Billion in 2015[1]—airport analysis has not lost any importance. How relevant security issues with these complicated system are, has been brought back to attention with the attacks on Brussels and Istanbul airports in 2016.

[1] According to http://data.worldbank.org/indicator/IS.AIR.PSGR.

G. Barbeito (✉) · M. Moll · S. Pickl · M. Zsifkovits
Universität der Bundeswehr München, Neubiberg, Germany
e-mail: gonzalo.barbeito@unibw.de

M. Moll
e-mail: maximilian.moll@unibw.de

S. Pickl
e-mail: stefan.pickl@unibw.de

M. Zsifkovits
e-mail: martin.zsifkovits@unibw.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_74

Ranging from microscopic models with focus on operational issues, to strategy-centered macroscopic models, airport modeling has been a popular research topic for the last two decades [8]. A well-known approach commonly used for mid to high level analysis is System Dynamics (SD) [1, 7]. This is a modeling and simulation technique introduced by Forrester in [3] for studying the behavior of systems over time with well documented models providing interesting insights for system improvement. SD models are basically composed of stocks, flows and delays. These simple elements allow being composed in such a way that complex systems can be represented [6]. In this paper we introduce a model for a generic airport using this technique. Instead of focusing on specific areas like boarding or security checks the emphasis is put on the interplay of the various subsystems of the airport as a whole. Being driven by the number of arriving and departing flights, it keeps track of the number of passengers at each point and the noise level of the airport. Following the idea of using SD for risk modeling and terroristic threat analysis in [2] our model includes emergency landings and heightened terror related threat levels based on random events. Our first results provide decision makers with a good and approachable overview of challenges and problems an airport can face. In order to achieve this, results can be either considered for each subsystem individually or aggregated for arbitrarily large parts of the model.

2 Model Description

The approach for this model assumes a pull logic in which the aircraft dynamics are determining the behavior of the rest of the system. Defining such a logic required the creation of an auxiliary time management submodel to align present events with future timetables.

The model breaks quite naturally into two major parts, the aircraft and the passenger dynamics, the first of which summarizes all dynamics and interactions directly related to the landing and takeoff procedures—both limited by structural and dynamical resources. Contrary to the structural resources, for which only the consequences can be seen, the dynamical ones are explicitly modeled through takeoff and landing availability and can be automatically reassigned matching the required load. Particular care had to be taken to keep track of passing hours and time slots correctly. The system for landing permissions works in the same fashion.

The other major part of the model consists of all subsystems relating to passenger flows. Its core is the submodel handling the load and analysis of the airport sections, which can be found in Fig. 1. The approach chosen for this model has the potential to cover all four objectives for analysis presented in [5] namely capacity planning, operational planning and design, security policy and planning, and airport performance review.

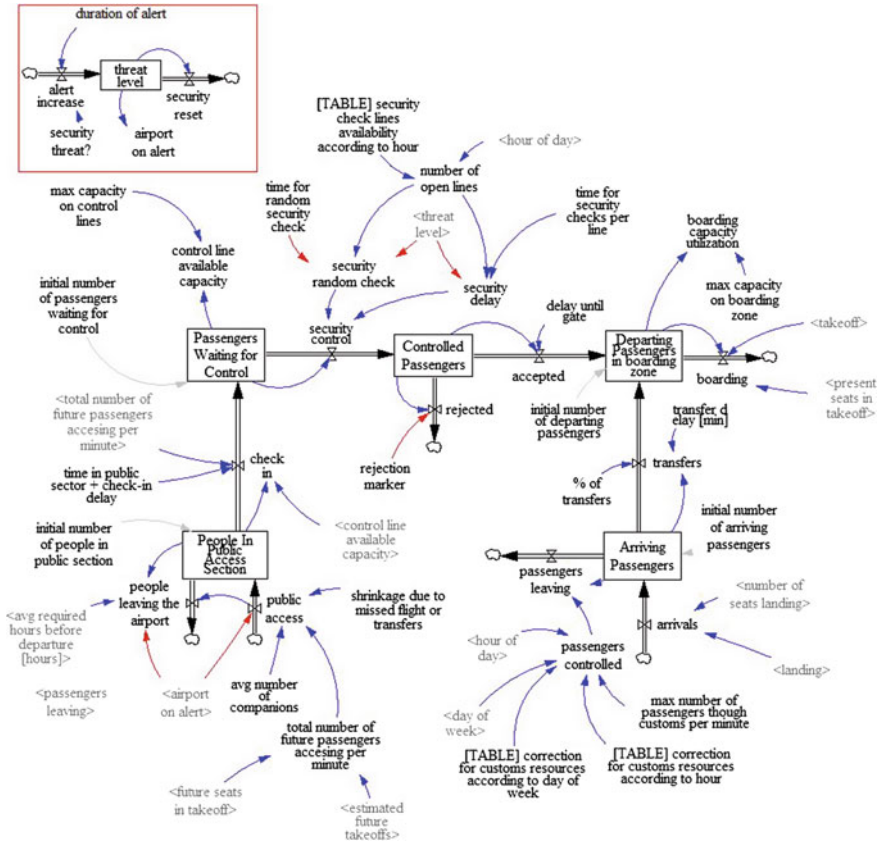


Fig. 1 Airport sections load and analysis

3 First Results

We now go on to describe simulation results obtained. First, results regarding typical operations are being discussed. These are followed up by observations in random extreme event scenarios.

Given a finite amount of resources, one of the managers highest priorities is to make the best possible use of them. Computational modeling in general and SD in particular has proven to be an effective approach for generating strategies to solve this problem to optimality [4]. As pointed out before, this model aggregates resources in two different groups, variable and structural. Variable resources account for all those susceptible of being modified in short periods of time, while fixed resources are modified in the medium to long term horizon. As a result of this distinction, policy optimization for the latter can be influenced only through improvements in

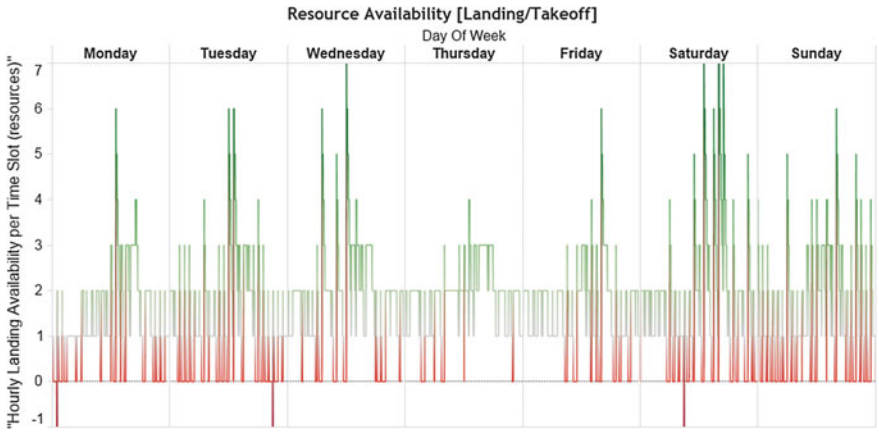


Fig. 2 Details for the available resources of the landing subsystem

the resource consumption schedule, while the former can be affected by resource allocation as well.

Figure 2 provides the results for current allocation and use of variable resources, particularly for airplane landing. The peaks are associated with wasted resources that could be potentially relocated to a time window with higher necessity for it. In the same way, pits indicate a profound scarcity. The results of the simulation on this level, show that the system is not well balanced, and needs to reconfigure either its resource allocation or its consumption schedule. The optimization of a policy will ideally influence both variables for a more balanced resource utilization.

The results for fixed resources are not as easily disaggregated. The basic restriction is set in place in the form of a minimum time between landings or takeoffs. Through this restriction the model accounts for several constrains, associated to structural characteristics of the airport, e.g. the number of available runways and the capacity of the airport for handling parked airplanes. Figure 3 provides a comprehensive detail on the airport load for the analyzed week. It is clear that the load is not correctly balanced, with several peaks that could be avoided.

As the aim of this work is not an accurate reproduction of a system, but to test the resilience of said system and its behavior under current conditions, we will highlight the changes of the system when put under stress. In Fig. 4 the effect of an emergency landing is shown. As can be seen, this increases the window between regular landings, resulting in delays for the queuing aircrafts and hence congestion of the passenger flow. Comparison of the two flows in Fig. 5 shows the effects of a terror threat on the passenger dynamics. It can be seen that longer security checks lead to empty boarding zones, as the passenger flows are stopped at this bottle neck.

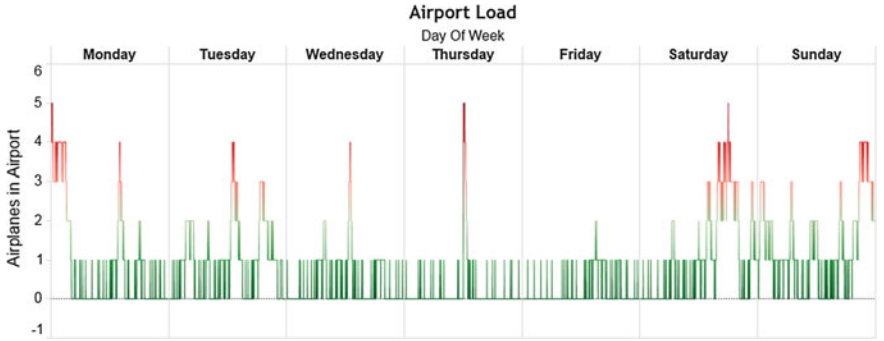


Fig. 3 System load measured in number of airplanes simultaneously at the airport

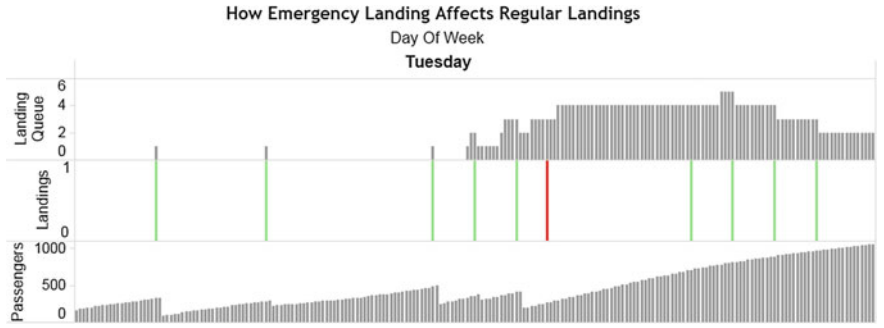


Fig. 4 Effects of an emergency landing—indicated by the *red line*—on queuing aircrafts and passengers after security controls

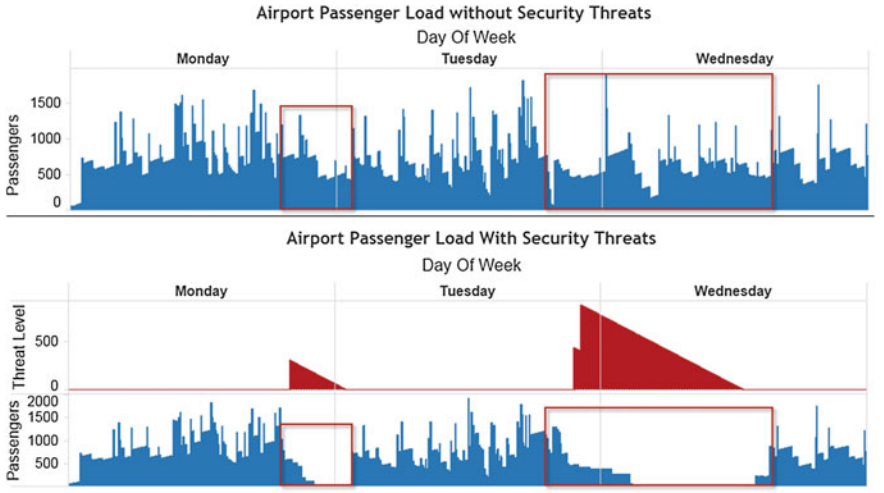


Fig. 5 Effect of security threats on passenger dynamics

4 Conclusion

In this paper we presented a new SD model for a generic airport. We also showed some first simulation results and the impact of security threats and emergency landings. The next step should be to adapt the model to match an existing airport. The changes for this should be mainly in the parameter tuning, as the macroscopic structure allows for a variety of different airport layouts. From this, policy optimisation can be applied to improve the use of structural and dynamical resources.

Acknowledgements We want to thank Andreas Tahedl, Michael Rampetsreiter, Jan-Peter Neutert and Martin Weiderer for constructing a first version of the model during their work with our group.

References

1. Bießlich, P., Schröder, M., Gollnick, V., Lütjens, K.: A system dynamics approach to airport modeling. In: 14th AIAA Aviation Technology, Integration, and Operations Conference, p. 2159 (2014)
2. Ezell, B.C., Bennett, S.P., Von Winterfeldt, D., Sokolowski, J., Collins, A.J.: Probabilistic risk analysis and terrorism risk. Risk Anal. **30**(4), 575–589 (2010)
3. Forrester, J.W.: Industrial Dynamics. M.I.T. Press, Cambridge (1961)
4. Grösser, S.N., Jovy, N.: Business model analysis using computational modeling: a strategy tool for exploration and decision-making. J. Manag. Control **27**(1), 61–88 (2016)
5. Manataki, I.E., Zografos, K.G.: A generic system dynamics based tool for airport terminal performance analysis. Transp. Res. Part C: Emerg. Technol. **17**(4), 428–443 (2009)
6. Sterman, J.D.: Business Dynamics: Systems Thinking and Modeling for a Complex World. Irwin/McGraw-Hill, Boston (2000)
7. Suryani, E., Chou, S., Chen, C.: Air passenger demand forecasting and passenger terminal capacity expansion: a system dynamics framework. Expert Syst. Appl. **37**(3), 2324–2339 (2010)
8. Wu, P.P., Mengersen, K.: A review of models and model usage scenarios for an airport complex system. Transp. Res. Part A: Policy Pract. **47**, 124–140 (2013)

Part XVIII
Simulation and Stochastic Modeling

Simulating the Diffusion of Competing Multi-generation Technologies: An Agent-Based Model and Its Application to the Consumer Computer Market in Germany

Markus Günther and Christian Stummer

Abstract Consumer adoption of innovations is a key concern for strategic management in many companies as adoption ultimately drives the market success of new products. The respective adoption processes are inherently complex due to the social systems (i.e., the respective consumer markets) from which they arise. Markets characterized by the simultaneous presence of several multi-generation technologies, wherein products that rest upon successively introduced generations of technology compete against each other, constitute a particularly challenging case. Our agent-based model contributes to the field of technology diffusion research in that it accounts for novel and advanced product features in each technology generation, the reluctance of (some) users to switch to a new (as yet unfamiliar) technology, and various social influences between consumers. Calibrated with data from several sources, our results closely replicate the actual development of the German consumer computer market from 1994 to 2013.

1 Introduction

Running an agent-based simulation can be as simple as instantiating an agent population, letting the agents behave and interact, and observing what happens globally [1]. Nevertheless, when studying complex systems like markets, societies, and organizations, such an approach can produce richer and more accurate results than traditional analytical approaches. Agent-based modeling and simulation has thus been widely adopted by researchers from various communities [15], particularly within the context of innovation and technology diffusion, where such an approach takes into account the heterogeneity of consumers who may differ in their individual preferences, behaviors, expertise, geographical position, etc. (for a review of agent-based

M. Günther · C. Stummer (✉)
Faculty of Business Administration and Economics, Bielefeld University,
Universitaetsstr. 25, 33615 Bielefeld, Germany
e-mail: christian.stummer@uni-bielefeld.de

M. Günther
e-mail: markus.guenther@uni-bielefeld.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_75

models of innovation diffusion see, for instance, [7], and for some recent applications see [3, 9, 11, 12, 14, 16]).

Today's continuous technological improvements give rise to successive generations of technology. Products belonging to a new generation usually offer specific innovative performance enhancements and/or new features, while the core functionality generally stays the same [8]. Such successive introductions may be rewarding not only for customers but also for companies (for an example see [4]). However, the specific patterns of adoption of each product generation are not yet well understood [13].

Our contribution lies at the intersection of these two streams of research: we propose an agent-based approach that simulates the market diffusion of several technologies as well as the diffusion of the products that rest upon the various technology generations over time. The underlying model accounts for (i) novel and/or advanced product features in each generation, (ii) interactions between multiple competing technologies, (iii) repeat and postponed purchase decisions, (iv) normative influences, and (v) a social network that reflects both spatial and social proximity between consumers. Its applicability is demonstrated by replicating the development of the German consumer computer market. A previous version of the model was presented at the Portland Conference for Management of Engineering and Technology [5].

The remainder of the paper is organized as follows: In Sect. 2, we outline our agent-based model. The sample application is then presented in Sect. 3. The paper concludes with a summary and an outlook to future research in Sect. 4.

2 The Agent-Based Model

An overview of the main entities and the model dynamics is depicted in Fig. 1 and is described in more detail in the following.

Products and Technologies Several (technological) generations of products are successively introduced into the market. Each product is characterized by various attributes that differ in their performances. A newer generation of a product may not only perform better than previously introduced products in regard to certain attributes (e.g., faster network connectivity), but may also have new, additional attributes (e.g., internet connectivity). Every product's attribute has a true performance value which, however, may not be instantly observable. Once consumers have adopted a product, they learn about its characteristics through first-hand experience. This experience may differ between consumers as well as by attribute. Every product also has a predefined price, which may vary over time, and a predefined date for its market introduction and its discontinuation. Finally, products belong to a particular technology, which enables us to capture diffusion patterns at both the product and the technology level.

Consumers Our agent-based model distinguishes all five phases of the adoption of an innovation as described by Rogers [10]. Initially, consumers are not aware of the

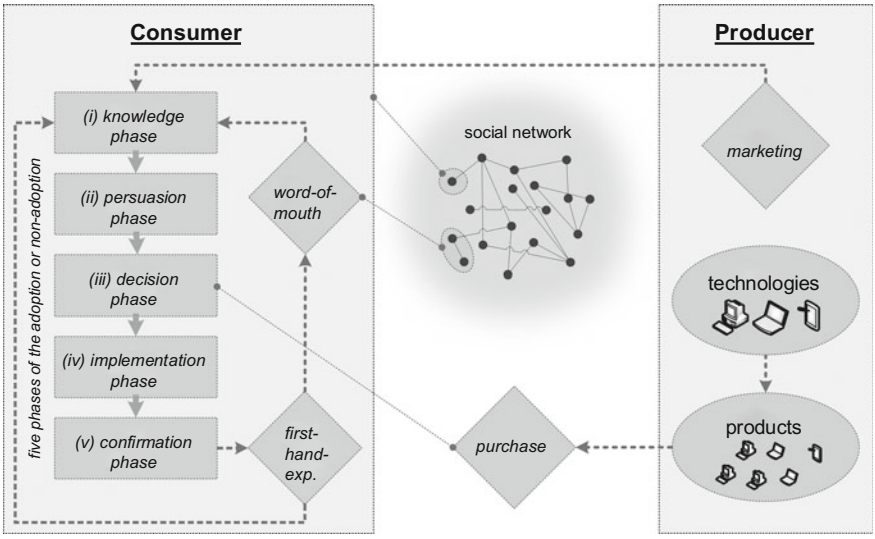


Fig. 1 Model overview

available products, their attributes, or their 'true' performances. In the course of the simulation presented here, consumers form their attitudes about products either by being exposed to marketing activities, through first-hand experience after purchasing, or by word-of-mouth from their peers. Note that word-of-mouth communication can be negative if personal experience with a product does not match the expectations.

In the current version of the model, consumers can only possess one product at a time and they replace it occasionally based on an individual buying cycle. If a purchasing need arises, they evaluate all available products of which they are aware. For the evaluation of a product, an additive utility function is used that takes into account (i) the attitudes of a consumer agent with respect to the available products and their attributes (analogous to [11]), (ii) social influence that may play a critical role in purchasing a product [2], and, of course, (iii) price. We also allow for re-purchases and postponed purchases. Reasons for the latter might be that consumers do not have sufficient information about a new product, that they are not aware of all (new) attributes of the product (as they have not heard about it via marketing activities or word-of-mouth), that their preference structure leads to a lower evaluation of the new product compared to the original product, or that its utility does not exceed a minimum utility threshold. After their purchase, consumers start to use the (new) product, thus learning more about its 'true' features (attributes).

Social Network When constructing the social network model, we followed the notion that agents with closer cognitive proximity (e.g., they are of the same consumer type) and/or geographic proximity have a higher probability of being interconnected. To this end, we have extended earlier approaches by [6, 11].

Marketing Marketing events make consumers aware of new products and/or attributes and provide consumers with information about the performances of (some) product attributes. Accordingly, each marketing event is characterized by the targeted product and topic (attribute) as well as the content (i.e., information about the performance, which might also be exaggerated in comparison to the true value). So far, our model implements only mass media advertising.

3 Application Case

Our model was implemented in AnyLogic 7.0.3. In order to demonstrate its application to the German (private) consumer market of desktop computers, notebooks, and tablets, we have relied on historical sales data that is publicly available through the 'Consumer Electronics Market Index Germany' (CEMIX) for the years 2005 through 2013. Further data was received for the years 1994 through 2004 from the same publisher through personal correspondence.

Parametrization Based on product characteristics such as computer performance (driven by, e.g., the processor type), mobility, battery life, and (internet) connectivity, we have formed several product generations for each of the considered technologies: desktop computer (seven generations), notebook (four generations), and tablet (one generation).

The required parameters for the consumer agents (preferences, communication behavior, and habits of adopting new products, etc.), the social network, the attributes of the considered products, and the effectiveness of marketing measures have been either derived from literature or calibrated to match the available sales data. Detailed information on the sources for the parameters are provided in [5].

Results Each simulation was initialized with 10,000 consumer agents and the time horizon was set to 20 years. Simulation runs were repeated fifty times using different seeds. A comparison between the empirical data and the outcome of the agent-based simulation is depicted in Fig. 2.

The outcome of the simulation experiment fits the actual market development for our application case exceptionally well—with the exception of small deviations in the last three years. Moreover, an in-depth analysis reveals that the measures that had to be set during simulation runs in order to achieve such a close fitting are plausible in that the resulting market behavior is consistent with common managerial experiences. For instance, it can be shown (for the simulated market) that a substantial increase in marketing effort is required to increase sales once consumers have formed opinions about a product's attributes. Compensating for a lack in technological performance with marketing is therefore viable only shortly after market introduction. We also found that the decline in sales of desktop computers can be explained by a combination of decreasing prices and new product characteristics within the competing technologies. Furthermore, novel attributes apparently function as an enabler for new technology generations and price has a strong impact on the sales rate

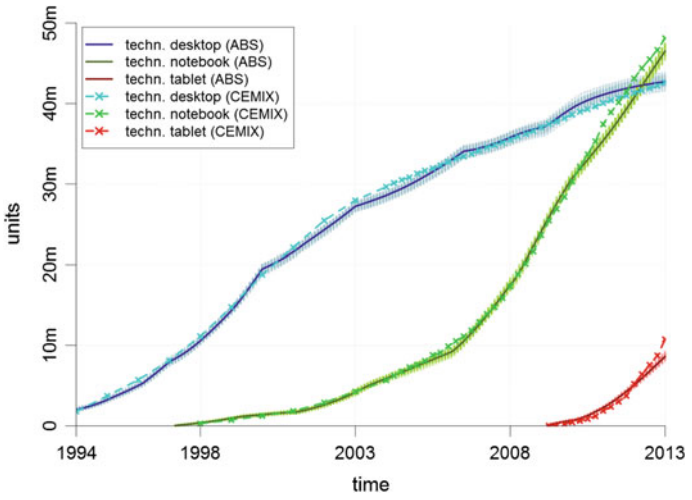


Fig. 2 Simulated sales versus real data

(as can be observed particularly well in the case of the diffusion of notebooks). Overall, the comparison of outcomes and market behavior of the simulated market with the real market provides additional evidence for the feasibility of agent-based models to properly represent such complex markets.

4 Conclusions

In this paper, we have described an agent-based model simulating the complex diffusion process of (competing) multi-generation technologies and have demonstrated its applicability by replicating the diffusion of three different computer technologies (desktop computer, notebook, and tablet) in Germany for the years 1994–2013.

Further research is possible in several directions. Once additional data becomes available, it will be interesting to investigate whether the small deviation between the real data and the simulation output concerning notebook technology is still present or whether this deviation is just an artifact. Next, the model could be extended by allowing for possessing more than one product simultaneously, which seems especially useful for models including tablets, as tablets are often used in addition to a desktop computer or notebook. Furthermore, the influence of teething problems (i.e., failure rate, bugs) on the diffusion process may be explored. Finally, the model might be tested for application in other fields (e.g., mobile phones).

Acknowledgements We would like to thank Immanuel Block for his support in acquiring the empirical data.

References

1. Axtell, R.L.: Why agents? On the varied motivations for agent computing in the social sciences. In: Macal, C.M., Sallach, D. (eds.) Proceedings of the Workshop on Agent Simulation: Applications, Models, and Tools, pp. 3–24. Argonne National Laboratory, Argonne (2000)
2. Delre, S.A., Jager, W., Bijmolt, T.H.A., Janssen, M.A.: Will it spread or not? The effects of social influences and network topology on innovation diffusion. J. Prod. Innov. Manag. **27**, 267–282 (2010)
3. Desmarchelier, B., Fang, E.S.: National culture and innovation diffusion: exloratory insights from agent-based modeling. Technol. Forecast. Soc. Change **105**, 121–128 (2016)
4. Druehl, C.T., Schmidt, G.M., Souza, G.C.: The optimal pace of product updates. Eur. J. Oper. Res. **192**, 621–633 (2009)
5. Günther, M.: Diffusion of multiple technology generations: an agent-based simulation approach. In: Kocaoglu, D.F., Anderson, T.R., Daim, T.U., Kozanoglu, D.C., Niwa, K., Perman, G. (eds.) Proceedings of the Portland International Conference for Management of Engineering and Technology (PICMET '16), pp. 2931–2940. PICMET, Portland (2016)
6. Günther, M., Stummer, C., Wakolbinger, L.M., Wildpaner, M.: An agent-based simulation approach for the new product diffusion of a novel biomass fuel. J. Oper. Res. Soc. **62**, 12–20 (2011)
7. Kiesling, E., Günther, M., Stummer, C., Wakolbinger, L.M.: Agent-based simulation of innovation diffusion: a review. Cent. Eur. J. Oper. Res. **20**, 183–230 (2012)
8. Kilicay-Ergin, N., Lin, C., Okudan, G.E.: Analysis of dynamic pricing scenarios for multiple-generation product lines. J. Syst. Sci. Syst. Eng. **24**, 107–129 (2015)
9. Palmer, J., Sorda, G., Madlener, R.: Modeling the diffusion of residential photovoltaic systems in Italy: an agent-based simulation. Technol. Forecast. Soc. Change **99**, 106–131 (2015)
10. Rogers, E.M.: Diffusion of Innovations, 5th edn. Free Press, New York (2003)
11. Stummer, C., Kiesling, E., Günther, M., Vetschera, R.: Innovation diffusion of repeat purchase products in a competitive market: an agent-based simulation approach. Eur. J. Oper. Res. **245**, 157–167 (2015)
12. Swinerd, C., McNaught, K.R.: Comparing a simulation model with various analytic models of the international diffusion of consumer technology. Technol. Forecast. Soc. Change **100**, 330–343 (2015)
13. van Rijnsoever, F.J., Oppewal, H.: Predicting early adoption of successive video player generations. Technol. Forecast. Soc. Change **79**, 558–569 (2012)
14. Xiao, Y., Han, J.: Forecasting new product diffusion with agent-based models. Technol. Forecast. Soc. Change **105**, 167–178 (2016)
15. Zhang, T., Siebers, P.-O., Aickelin, U.: Simulating user learning in authoritative technology adoption: an agent based model for council-led smart meter deployment planning in the UK. Technol. Forecast. Soc. Change **106**, 74–84 (2016)
16. Zsifkovits, M., Günther, M.: Simulating resistances in innovation diffusion over multiple generations: an agent-based approach for fuel-cell vehicles. Cent. Eur. J. Oper. Res. **23**, 501–522 (2015)

Decomposition of Open Queueing Networks with Batch Service

Wiebke Klünder

Abstract The decomposition method for non-product form networks with non-exponentially distributed interarrival and service times assumes that nodes within the network can be treated being stochastically independent and internal flows can be approximated by renewal processes. The method consists of three phases to calculate the interarrival times of a node: merging, flow, splitting. Some well-known approximation formulas for ordinary single class open queueing networks calculate the characteristics in each phase for each node as shown by Kuehn, Chylla, Whitt and Pujolle/Ai. Node performance measures such as mean queue length are determined by using approximation formulas for non-Markovian queues. In 2011 the decomposition method was extended to open queueing networks with batch processing using the approximation formula described by Pujolle/Ai. A comparison with discrete event simulation as benchmark shows that the approach provides good results. Thus, the approach was expanded for the approximation given by Kuehn, Chylla and Whitt. Since the method consists of several phases it is possible to combine different formulas. For example, merging will be approximated by Kuehn and flow by Whitt. To perform an evaluation the benchmark was done in regard to the 2011 publication. Approximation formulas with the same approach generate similar results. In some cases, it is apparent that some formulas have advantages over other ones and a few tend to larger errors. Thus, the focus of interest particularly addresses the load and batch size changes within the network and the impact on the accuracy of the decomposition method as a fast solver or pre-evaluation for optimization using simulation.

1 Introduction

The importance of analysis of non-product form networks by applying approximations has increased steadily in recent years. The most important strategy approach is given by the decomposition method. The decomposition method enables an

W. Klünder (✉)
Simulationswissenschaftliches Zentrum Clausthal-Göttingen,
Arnold-Sommerfeld-Straße 6, Clausthal-Zellerfeld, Germany
e-mail: wiebke.kluender@tu-clausthal.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_76

isolated treatment of the nodes within the network. The method is particularly applied in planning and optimization of production systems by calculating characteristics of each node. In this paper, a decomposition method will be presented serving primarily as a pre-evaluation tool. If the calculated characteristics move in acceptable ranges Monte-Carlo simulations can be performed.

Until now the decomposition method for open queuing networks with batch service was developed using the approach of Pujolle/Ai [1]. The aim is to expand the method to common approximations developing the approaches to batch service and to transfer them to the developed method of [1]. This includes the approximate formation of the superposition of the input streams by Kühn [2] and Chylla [3] as well as the approximation of the departure stream by Whitt [4], Kühn and Chylla.

2 Description of the Model

The open network consists of 1 to N nodes numbering successively and presenting $GI^{X_i}/GI^{(b_i,b_i)}/c_i$ queueing systems. Jobs arrive in groups of size b_0 from outside the network according to a renewal process with rate $\lambda_0 < \infty$ and the squared coefficient of variation $SCV[I_0] < \infty$. $0 \le p_{ij} \le 1$ describes the transition probability that an arriving batch reaches node j from node i and $\sum_{i=1}^{N} p_{0i} = 1$ applies meaning jobs enters the network from outside. Each queueing systems have c_i identical servers, an unlimited waiting room and the FCFS queueing discipline. The service starts if a batch of the required size b_i was generated. The service times are distributed as some random variables S_i with rates $\mu_i < \infty$ and $SCV[S_i]$. It is assumed that the interarrival and service times are independent. After a complete service of a batch it will arrive in a form of a batch to the subsequent node according to the transition probabilities. X_i is described by an integer random variable and represents the input size of the groups at node i. The first and second moment are calculated by

$$E[X_i] = \frac{\sum_{j=0}^{N} b_j \cdot \tau_j \cdot p_{ji}}{\sum_{j=0}^{N} \tau_j \cdot p_{ji}} \qquad E[X_i^2] = \frac{\sum_{j=0}^{N} b_j^2 \cdot \tau_j \cdot p_{ji}}{\sum_{j=0}^{N} \tau_j \cdot p_{ji}}.$$

τ_i denotes the relative throughput and can be determined by a modified traffic equation:

$$\tau_i = p_{01} \cdot \frac{b_0}{b_i} + \sum_{j=1}^{N} \tau_j \cdot p_{ji} \cdot \frac{b_j}{b_i} \qquad \tau_0 := 1.$$

The modified arrival rate of batches can be calculated by $\lambda_i^* = \lambda_0 \cdot \tau_i$ and the modified utilization by $\rho_i^* = \lambda_i^*/(c_i\mu_i) < 1$.

3 Decomposition of Open Queueing Networks with Batch Service

3.1 Phase 1: Merging

There exist three different approaches to form the superpositions of the arrival streams. The first approach was developed by Pujolle/Ai [5]. The counting process representing the incoming jobs and incoming groups, respectively is described by knowledge of the asymptotic behavior of renewal processes:

$$SCV[I_i] = \left(\sum_{j=0}^{N} \tau_j p_{ji} \right)^{-1} \sum_{j=0}^{N} \tau_j \cdot p_{ji} \cdot SCV[A_{ji}]$$

Chylla uses the approach in order to approximate the splitting of the departure stream (see phase 3):

$$SCV[I_i] = 1 + \sum_{j=0}^{N} \frac{\lambda_j^*}{\lambda_i^*} p_{ji} (SCV[A_{ji}] - 1).$$

The approach of Kühn is based on a case-by-case analysis which depends on the values of $SCV[A_{ji}]$ (see phase 3):

$$SCV[I_i] = 2 \cdot \frac{t_1 + t_2}{(t_1 \cdot t_2)^2} \cdot (I^1 + I^2 + I^3 + I^4) \qquad t_j = \frac{1}{p_{ji}\tau_j}, j = 1, 2.$$

The components $I^1, \ldots, I^4$ are either a composition of hypoexponentially, hyperexponentially distributed sub-processes or a mixture. For details see [2].

After the interarrival times of the single jobs has been determined the interarrival times of batches will be approximated by [1]:

$$SCV[I_i^*] \approx \frac{E[X_i]}{b_i} (SCV[X_i] + SCV[I_i]).$$

3.2 Phase 2: Flow

There are fundamentally two approaches to approximate the departure stream in a non-product form network. Pujolle/Ai and Chylla use the approach

$$D_i = \begin{cases} S_i & : \textit{with probability} \quad \rho_i^* \\ S_i + I_i^* & : \textit{with probability} \quad 1 - \rho_i^* \end{cases}$$

and it results for Pujolle/Ai according to the calculation of the first and second moment of the process D_i

$$SCV[D_i] \approx \rho_i^{*2} SCV[S_i] + (1 - \rho_i^*) SCV[I_i^*] + \rho_i^* (1 - \rho_i^*)$$

and a slightly modified version of Chylla

$$SCV[D_i] = 1 + P_i^2 (SCV[S_i] - 1) + (1 - P_i)(SCV[I_i^*] - 1),$$

where P_i is described by the Erlang-C formula. Whitt and Kühn use the approach of Marshall [6] to approximate the departure stream basing on Lindley's recursion of waiting times. The formula

$$SCV[D_i] \approx 1 + (1 - \rho_i^{*2}) \cdot (SCV[I_i^*] - 1) + \frac{\rho_i^{*2}}{\sqrt{c_i}} \cdot (SCV[S_i] - 1)$$

represents the approximation of Whitt and Kühn developed the approximation

$$SCV[D_i] = SCV[I_i^*] + 2\rho_i^{*2} SCV[S_i] - \rho_i^{*2} (SCV[I_i^*] + SCV[S_i]) g_{KLB},$$

where g_{KLB} is the correction factor given by Krämer/Langenbach-Belz [7].

3.3 Phase 3: Splitting

The splitting of the departure stream in accordance with the transition probabilities can be considered as a Bernoulli experiment. After service completion at node i, jobs are directed to node j with probability p_{ij} and with probability $1 - p_{ij}$ they are routed elsewhere. The number of the first batch to be directed to node j is geometrically distributed. The first moment and the variances of the splitting process are calculated by using the Wald's equation respectively the Blackwell-Girshick equation. The squared coefficient of variation results by $SCV[\cdot] = (E[\cdot^2]/E[\cdot]^2) - 1$:

$$SCV[A_{ij}] = 1 + p_{ij}(SCV[D_i] - 1).$$

If the phases are inserted successively into each other a system of linear equations is formed whose solutions provide the squared coefficient of variation of the interarrival times of the batches. Characteristics like the average number of individual jobs in the system of the various queueing systems can be determined by the modified formula of Allen-Cunneen [1] and the correction factor of Krämer/Langenbach-Belz:

$$E[N_i] \approx E[Z_{\infty,i}] + b_i \cdot E[Q]_{GI/GI/c} g_{KLB}(\rho_i^*, SCV[I_i^*], SCV[S_i]) + b_i c_i \rho_i^* + h.$$

4 Numerical Results

Due to the independence of the phases, the presented approaches can be arbitrarily combined, e.g. merging will be approximated by Kühn and flow by Whitt. The benchmark which was done in regard to [1] was used to evaluate the decomposition method using all possible combinations of the presented approximation approaches. The Fig. 1 shows the reference network. All in all, 16 cases were investigated in detail differing in the characteristics of the utilization, batch sizes, number of servers and $SCV[S_i]$.

Exemplarily, case 11 (benchmark: table 3, case 3) will be evaluated shortly. Table 1 presents the parameterization of the open queueing network. Table 2 summarizes the results and shows the relative errors. The relative error is the discrepancy between calculated approximate values by the decomposition method and the mean of the simulation results. The ratio of mean input batch size ($E[X_4] = 6.048$) and batch size b_4 at node 4 explains the increased discrepancies. A similar phenomenon occurs at the node 2 ($E[X_2] = 3 > b_2$). These situations affects an overestimation of the $SCV[I_i^*]$ and at last of the approximate characteristics. The approximations of the input stream of Pujolle/Ai and Kühn are robust. In contrast Chylla's approximation caused larger errors at node 2. The study of this case also clearly shows that Chylla's approximation combined with the approximation of the departure stream based on Marshall does not work well if there exist large changes of batch size (node 3). The formation of the superposition of the arrival streams under the circumstances of larger batch size changes revealed weaknesses of the approximation from Kühn (node 4).

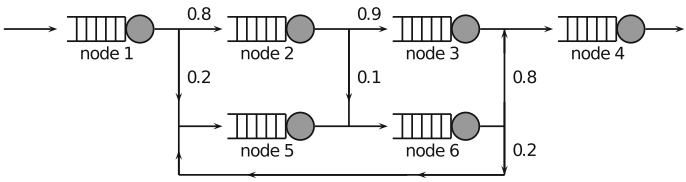


Fig. 1 Reference model

Table 1 Queueing network with $\lambda_0 = 2$, $SCV[I_0] = 1$, $b_0 = 1$, $E[S_1] = 3.6$, $E[S_2] = 0.45$, $E[S_3] = 4.5$, $E[S_4] = 4$, $E[S_5] = 8.889$ and $E[S_6] = 11.4$

Node	b_i	c_i	ρ_i	$SCV[S_i]$
1	3	3	0.8	0.25
2	1	1	0.72	0.25
3	10	1	0.648	0.25
4	5	2	0.8	0.25
5	2	4	0.6	0.25
6	3	3	0.887	0.25

Table 2 PA = Pujolle/Ai, W = Whitt, C = Chylla, K = Kühn. 1st position: Merging, 2nd position: flow

Node	Error in %											
	PA-PA	PA-W	PA-C	PA-K	C-PA	C-W	C-C	C-K	K-PA	K-W	K-C	K-K
1	1.027	1.027	1.027	1.027	1.027	1.027	1.027	1.027	1.027	1.027	1.027	1.027
2	2.291	8.103	6.154	0.889	26.667	28.547	27.897	25.607	2.291	8.103	6.154	0.889
3	2.893	1.876	2.695	2.505	0.099	19.468	0.637	25.337	2.893	1.876	2.695	2.505
4	25.956	26.747	28.631	21.96	31.919	36.011	34.14	32.751	31.369	31.499	32.861	29.271
5	3.745	3.55	3.631	3.826	3.437	3.307	3.355	3.485	3.712	3.534	3.615	3.777
6	11.485	6.827	5.399	9.601	12.017	4.103	1.977	7.525	11.43	7.464	6.728	9.502

5 Conclusions

All approaches yield acceptable results being useful as pre-evaluation for optimization. Generally it has been shown that the approximation of the input streams from Pujolle/Ai and Kühn are robust. The approximate approach from Chylla on the other hand caused in cases of larger batch size changes high errors (cases 9–16, benchmark: tables 3 and 4). The approach of Kühn, who has a complex case distinction is more difficult to handle than the approach of Pujolle/Ai.

The two approaches of the approximation of the departure streams yield similar results which could be expected since the approaches provide similar approximations. An interesting phenomenon discovered in many cases is that if $b_i < E[X_i]$ and $SCV[S_i] \to 0$ the approach based on Marshall's method works better than the approach from Pujolle/Ai and in case of $b_i > E[X_i]$ and $SCV[S_i] \to 0$ Pujolle/Ai's approach provides better approximations than the approach of Marshall. In the application it is possible to make a case analysis for each node to approximate the departure streams to reduce the error of the decomposition method.

References

1. Hanschke, Th., Zisgen, H.: Queueing networks with batch service. Eur. J. Ind. Eng. **5**(3), 313–326 (2011)
2. Kuehn, P. J.: Approximate analysis of general queuing networks by decomposition. IEEE Trans. Commun. **27**(1), 113–126 (1979)
3. Chylla, P.: Zur Modellierung und approximativen Leistungsanalyse von Vielteilnehmer-Rechensystemen. Dissertation, TU München (1986)
4. Whitt, W.: The queueing network analyzer. Bell Sys. Tech. J. **62**(9), 2779–2815 (1983)
5. Pujolle, G., Ai, W.: A solution for multiserver and multiclass open queueing networks. INFOR **24**(3), 221–230 (1986)
6. Marshall, K.T.: Some inequalities in queuing. Oper. Res. **16**(3), 651–668 (1968)
7. Krämer, W., Langenbach-Belz, M.: Approximate formulae for the delay in the queueing system gi/g/1. In: Proceedings of the 8th International Teletraffic Congress (ITC 8), pp. 235/1–235/8 (1976)

Decision Support for Power Plant Shift Configuration Using Stochastic Simulation

Pia Mareike Steenweg, Matthias Schacht and Brigitte Werners

Abstract Power generation companies have to ensure a secure supply of power to their customers at any time. Hence, this particular application context implies a special feature of shift planning where a very robust solution of assignment is needed. In daily business this means all business functions have to be staffed competently at any time, otherwise a smooth power plant operation cannot take place. In this regard, optimal shift assignment is a highly important and complex task, where reliability is prioritized with respect to all other criteria, e.g. employee's interests. In order to test a shift configuration on operational level, we conceptualise a reactive framework to support tactical decision making. The concept switches between optimisation and stochastic simulation which takes uncertainty associated with employee sickness into account. An exemplary case study with realistic data of a power generation company analyses the operational consequences of uncertain absences on the performance of a given shift configuration.

1 Introduction

A stable power supply is extremely important so that it is even regulated by law. Therefore, power generation companies have to ensure a smooth operation, which bases on the working machinery as well as on enough and qualified manpower. The latter can be influenced by appropriate shift planning, which comprises a variety of business functions to cover. Each business functions has to be staffed at any time to guarantee a smooth operation. Hence, a very robust shift configuration is needed. In this context, absences of the workforce due to sickness is a crucial aspect since it possibly causes under-staffing or unfavourable assignments. However, if only these aspects were considered, the worker's preferences would not be respected. The workers have individual and shared interests like fairness within the workforce or free

P.M. Steenweg (✉) · M. Schacht · B. Werners
Chair for Management, esp. Operations Research and Accounting,
Faculty of Management and Economics, Ruhr University Bochum, Bochum, Germany
e-mail: pia.steenweg@rub.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_77

week-ends. Thus, the shift planner has to satisfy both the company's and the worker's interest when deciding on the tactical shift configuration.

This contribution analyses the effects of uncertain absence on the performance of a given shift configuration. While the assignment reliability has to be assured at any time, results focus on employee's objectives. Thus, the special decision situation is analysed before a reactive framework modelling the real-world decision process is developed. Section 4 presents first results of an exemplary case study and Sect. 5 gives a short conclusion and an outlook.

2 Related Literature and Decision Analysis

Qualified personnel planning requires a good understanding of the related consequences and is highly important in power generation companies as stable operation is the essential objective. The short-term planning for example includes the suitable assignment of attendant employees to business functions. Unexpected notification of sick workers as stated in [1] complicates the decision situation as well as the dependence of taken strategical and tactical decisions. Therefore, staffing in the long-term has to ensure that enough qualified workers are available and the scheduling in the mid-term has to schedule them equally over the shift groups. Thus, the preceding interdependences demonstrate the importance of an integrated approach including the consequences of the decision on lower planning levels as also shown in [6].

This contribution analyses the shift configuration for a power generation company in-depth, which has two special challenges: First, many different functions have to be covered; second, a very robust solution is needed where all tasks are fulfilled at any time. The latter might lead to expensive over-staffing as shown in [3]. Note that the problem of over- and under-staffing has been widely addressed in recent literature [5]. To comply with these challenges, we analyse shift configurations on a tactical level by prioritising the consequences on the robustness on the operational level. Sickness arises randomly and for an unpredictable time. Therefore, we examine the influence of unexpected sickness while taking predictable absence, like holiday and training, into account.

In this context, the non-absent workers of the scheduled shift group are assigned optimally to the various functions that need to be covered by every shift group. Although every worker has a primary function he or she usually covers, adjustments regarding the workers' skills are regularly needed due to unexpected absences as stated in [2]. Hence at the beginning of each shift, the foreman reallocates the attendant workers to cover any function as qualified as possible. It is not unlikely that a shift is overstaffed since a smooth operation has the highest priority and a breakdown of operations due to a missing personnel is unacceptable. In practice, employees who are not assigned to a specific function will still be available. They carry out maintenance or other accompanying tasks or do trainings-on-the-job to appropriately comply with their contractual workload. Consequently, an subordinated objective is

a balanced assignment of workers to tasks which is favourable if at the same time assignment robustness stays the same.

Our proposed concept covers an optimal assignment incorporating the shift foreman's approach by respecting the company's interest of *reliability* to guarantee a stable power generation process. Additionally, employee's interest like *fairness* in the actual workload is considered as far as possible.

3 A Reactive Optimisation Framework for Decision Support

As mentioned in Sect. 2, the daily manpower depends on predictable (e.g. holidays) and unpredictable (e.g. sickness) employee attendance. In the following, we introduce a reactive concept to support the decision makers by examining the performance of varying shift configurations under uncertainty. For this purpose we assess the employee's attendance taking predictable absences (holidays and trainings) for T shifts into account. Furthermore, we simulate sickness for the current shift $\tau \in \{1, \ldots, T\}$. Based on the resulting attendance for shift τ and the anticipated for future shifts, each of the attendant workers i is optimally assigned to a function j for the current shift τ and planned for the following shifts with $t \in \{\tau + 1, \ldots, T\}$. The assignment of the past shifts with $t \in \{1, \ldots, \tau - 1\}$ is fixed as it cannot be changed any more. The interaction between past, current and future assignments can be seen in Fig. 1. As unpredicted changes in absence can occur every day, the simulation and optimisation is rerun for every $\tau \in \{1, \ldots, T\}$. Thus, the proposed concept has a reactive structure and conceptualises the assignment process as realistic as possible.

The optimal assignment of workers to functions is determined by an integer optimisation model of which the most relevant elements will be presented below:

$$\min \quad \sum_{j \in \mathcal{J}} \sum_{t \in T} G_j m_{jt} + \varepsilon\, n \tag{1}$$

The objective function (1) minimizes the number of un-covered functions by the sum of under-staffing m_{jt} over all functions and periods, weighted by G_j which considers the varying impact of different functions. If a complete schedule without under-staffing in any shift is assigned, $\sum_{j \in \mathcal{J}} \sum_{t \in T} m_{jt}$ equals zero. By the first part of the objective function, company's objective is taken into account. Additionally, the

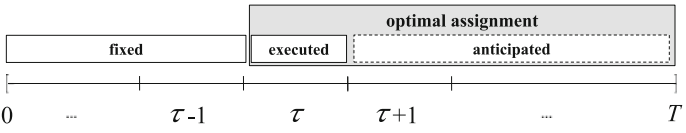


Fig. 1 Interaction between current and past assignments of workers to functions

objective function considers employee fairness by n, which denotes the maximum number of shifts a worker is assigned to a specific function. Thus, the workload is balanced by avoiding situations where some workers are extremely frequent and others rather seldom assigned. To ensure a stable power supply, generating a complete schedule has the highest priority. As a result, n is only weighted with a marginal term ε. Consequently, the assignment is optimised in terms of assigning qualified workers to all functions and provided that several assignments are optimal, the best assignment with respect to employee's interests is determined as discussed in [4].

$$\sum_{j \in \mathscr{J}} \sum_{t \in T} c_{ijt} \leq n \quad \forall\, i \in \mathscr{I} \tag{2}$$

Constraints (2) determine the number of shifts every employee is assigned to a specific function. The binary assignment variable $c_{ijt} \in \{0; 1\}$ equals 1 if worker i is assigned to function j in shift t, and zero otherwise. The minimax value n ensures choosing the number of shifts n of those worker i who is assigned most frequently. Due to the importance of a robust and smooth operation as mentioned in Sect. 2, there might be overstaffed shifts which will lead to a situation where a worker is present but will not be assigned to a specific task (i.e. $c_{ijt} = 0$). However, the worker will be able to support in general work.

The presented parts of the optimisation model reflect the dynamic relationship of the components *reliability* and *fairness*. On the one hand, m_{jt} representing company's interest is re-optimised in every shift independent of prior or future shifts. On the other hand, n is determined over all shifts of the planning horizon, which includes the fixed past assignments as well as the anticipated future ones. Therefore, n indicates the impact of the reactive structure considering current and future shifts while respecting the fixed assignments in the past. Within this concept, a given tactical shift configuration can be tested on an operational level, where an optimal assignment takes place (especially taking non-anticipated uncertain absences on a day-to-day basis into account). The concept will be tested in Sect. 4 with respect to its suitability for real-world application.

4 Evaluating Fairness in the Reactive Optimisation

In this section, the presented concept is applied using realistic data from a German power generation company with a given configuration consisting of 4 fixed shift groups working daily in 3 shifts of 8 h. Employee fairness from a tactical point of view is guaranteed by rotating every two days (except for Sundays) from early to late to night to free shift (see Table 1). Thus this roster complies with labour law, e.g. days-off and rests between two shifts. As a result of strategic planning, each shift group is sufficiently staffed and qualified to balance the historically expected values of absences. A time period of 90 days is considered in which a maximum of 66 shifts per worker occurs according to the roster in Table 1. This maximum is individually

Table 1 Roster with 4 groups

Day	Early	Late	Night	Free
Mo	1	2	3	4
Tu	1	2	3	4
We	4	1	2	3
Th	4	1	2	3

deducted by holidays, sickness etc. With an uniform distribution of these days, every worker would be assigned in 43.37 shifts (ideal but unrealistic solution).

The given situation is evaluated in terms of the assignment criteria *reliability* and *fairness*. Therefore, we compare the results of the introduced reactive framework to scenario-optimal results, which occur if sickness is known in advance for all shifts. The scenario-optimum represents the lower bound for maximum assignments since the objective function is minimised.

The reactive framework re-optimises the assignment for every shift τ based on updated information on sickness. Since the prioritised objective *reliability* is not affected by past and future assignments, the results of reliability coincide with the scenario-optimal assignment. Consequently, the optimal reactive assignments are as robust as possible from a company's perspective which renders unnecessary further analysis. Therefore, this analysis focuses on worker *fairness*, particularly on how uncertain absences affects employee's interests of a fair assignment to functions.

Analysing the maximum number of shifts a worker is assigned relates to the reactive structure. Since it is linked to the fixed assignments in the past, *fairness* in terms of maximum workload can only be optimised for the current and future shifts. The past assignments reduce the degrees of freedom in the process of determining a fair assignment. Figure 2 illustrates the maximum number of assigned shifts n for the impossible to obtain ideal solution (dashed) and reactive framework (black) and the scenario-optimal results (gray) by distribution functions for 50 simulation runs.

In Fig. 2, the distribution function of the reactive concept is about 2 maximum assigned shifts more to the right than the scenario optimum. This is tantamount to about 2 shifts ($\approx 4\%$) the worker with most assignments has to fulfil additionally

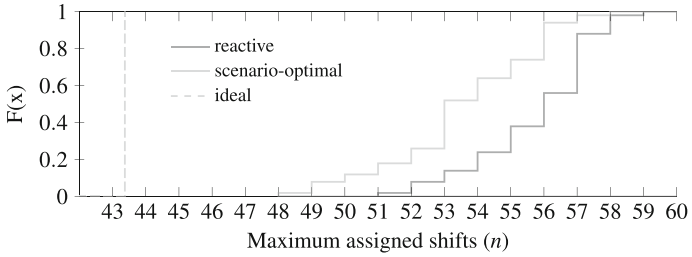


Fig. 2 Distribution functions of fairness for reactive, scenario-optimal and ideal

in 90 days in the reactive concept. In relation to the ideal solution with uniformly distributed absence (dashed) of 43.37 shifts, this gap of about 2 shifts appears less significant. The preceding analysis has evaluated the performance of the reactive framework compared to the scenario-optimal results. In terms of *reliability*, the reactive framework coincides with the scenario-optimal results. Considering *fairness*, the reactive concept performs very well even for unexpected changes in attendance.

5 Conclusion and Outlook

We have presented a general concept for decision support on a tactical level to test given shift configurations in which a reactive framework switches between simulation and optimisation. In particular, we have analysed a situation in which an extremely robust and smooth operation is prioritised over employee fairness. The particular application requires a robust shift configuration since an under-staffing of functions prevents a smooth operation. Moreover all workers are assumed to be available in their scheduled shifts. However, in case of over-staffing, the unassigned workers will do general work, which is fairly assigned by the reactive framework.

The presented concept has been analysed using realistic data of a German power generation company. Hence it can be used to optimise current and future assignments for every shift, whereas past assignments are fixed. The analysis of the reactive framework yields scenario-optimal results concerning the company's interest. Regarding the employee's objectives, the results are almost as good as the scenario-optimum under certainty as they are subordinated in the optimisation.

In conclusion, the reactive framework provides high quality decision support for real-world applications. Further research will be to apply the presented concept on further shift configurations to compare performance of different configurations with respect to the proposed criteria.

References

1. Bard, J.F., Purnomo, H.W.: Hospital-wide reactive scheduling of nurses with preference considerations. IIE Trans. **37**(7), 589–608 (2005)
2. De Bruecker, P., van den Bergh, J., Beliën, J., Demeulemeester, E.: Workforce planning incorporating skills: state of the art. Eur. J. Oper. Res. **243**(1), 1–16 (2015)
3. Ingels, J., Maenhout, B.: The impact of reserve duties on the robustness of a personnel shift roster: an empirical investigation. Comput. Oper. Res. **61**, 153–169 (2015)
4. Prot, D., Lapègue, T., Bellenguez-Morineau, O.: A two-phase method for the shift design and personnel task scheduling problem with equity objective. Int. J. Prod. Res. **53**(24), 7286–7298 (2015)
5. van den Bergh, J., Beliën, J., De Bruecker, P., Demeulemeester, E., De Boeck, L.: Personnel scheduling: a literature review. Eur. J. Oper. Res. **226**(3), 367–385 (2013)
6. van der Veen, E., Hans, E.W., Post, G.F., Veltman, B.: Shift rostering using decomposition: assign weekend shifts first. J. Sched. **18**(1), 29–43 (2015)

Part XIX
Software and Modeling Systems

Planarization of CityGML Models Using a Linear Program

Steffen Goebbels, Regina Pohle-Fröhlich and Jochen Rethmann

Abstract CityGML is an XML based description standard for 3D city models that requires model buildings to have planar roof facets. Unfortunately, current tools that generate building models from airborne laser scanning point clouds violate this requirement to some extent. We propose a definition of approximate planarity and present a post-processing tool that establishes approximate planarity using linear optimization. It preserves the characteristic shape of roofs. In most cases, triangulation of non-planar facets is no longer needed to heal models. This not only reduces the number of facets and increases performance of applications that process city models, but also avoids disturbing edges in 3D prints.

1 Introduction

CityGML is an established, XML-based standard for describing and exchanging virtual 3D city models. Such models can be used for planning, simulation, and marketing, cf. [8]. Still challenging is the automated generation of building models from cadastral data in combination with sparse point clouds from airborne laser scanning. There are two different algorithmic approaches: Model driven methods fit standard roof shapes with a point cloud. Data driven methods try to detect single planar facets and combine them to complete roofs (cf. [6, 7, 11]). To make facets fit together, small adjustments might be necessary. This often leads to a violation of the CityGML requirement of roof facet planarity [5, p. 25], cf. [12]. The problem is more relevant for data driven methods, because real roof facets often are not exactly planar, especially if buildings are old. For example, the algorithm of [4] merges vertices

S. Goebbels (✉) · R. Pohle-Fröhlich · J. Rethmann
Faculty of Electrical Engineering and Computer Science, Niederrhein University of Applied Sciences, 47805 Krefeld, Germany
e-mail: steffen.goebbels@hs-niederrhein.de

R. Pohle-Fröhlich
e-mail: regina.pohle@hs-niederrhein.de

J. Rethmann
e-mail: jochen.rethmann@hs-niederrhein.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_78

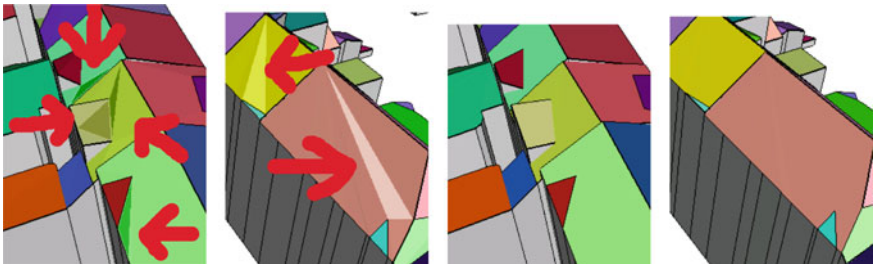


Fig. 1 Triangulation of non-planar roofs leads to visible triangles with different normal vectors. *Left* two pictures show the original model, *right* pictures display the result of optimization

with equal x- and y-coordinates, and z-coordinates within a certain range determined by a threshold value. This deliberately violates planarity. Missing planarity seems to be not only a problem of data driven methods. The current model driven North Rhine Westphalian city model [10] also shows a few non planar roof facets, probably because of rounding errors, further examples can be found in [1]. The problem can be solved using tessellation. But by splitting up non-planar surfaces into triangles, the number of roof facets increases and single triangles might become visible because they do not fit with the building's geometry, see Fig. 1. Alam et al. [1] propose an iterative least-squares fitting algorithm, where for each roof facet a plane is computed that fits the z-coordinates of the polygon's vertices best according to a square norm. This has to be done iteratively, because local adjustment of one polygon might undo changes to polygons previously dealt with. Therefore, the algorithm stops after a maximum number of iterations and does not assure overall optimal results. This is our motivation to formulate a global optimization problem. Optimization of energy functions is a standard tool in architecture reconstruction. For example Arikan et al. [2] use a Gauss-Seidel algorithm to snap together polygons within a semi-automated framework. However, the linear structure of planes suggests the use of linear programming. Mixed integer linear programming is an established means for surface reconstruction using optimum binary labelings, see for example [3] and the literature cited there. In contrast to these approaches that lead to a surface model, we start with an existing model and improve it.

2 Linear Program

City models are expected to be given in level of detail 2 (LoD2), i.e. they consist of wall and roof polygons. We define a roof topology T as the given set of roof polygons P_k, $k \in [n] := \{1, 2, \ldots, n\}$. Each polygon is a vector of at least three different vertices $p_{k,1}, \ldots, p_{k,m_k} \in V$, $p_{k,j} = (p_{k,j}.x, p_{k,j}.y, p_{k,j}.z)$, where $V \subset \mathbb{R}^3$ is the roof's finite vertex set. The edges between subsequent polygon vertices and between the last and the first vertex define the boundary of a roof's surface facet. Vertices $p \in V$ can be

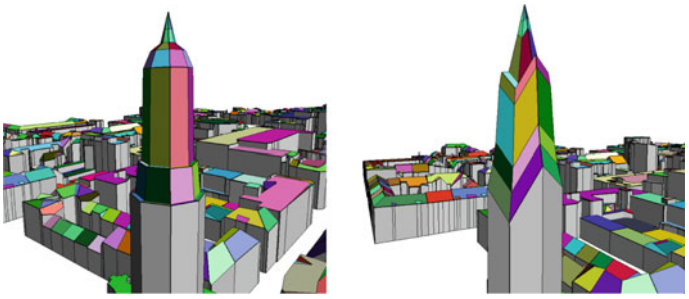


Fig. 2 From *left* to *right*: Planarization was done to the tower's nearly vertical roof facets without considering deviation of x- and y-coordinates, i.e. with $\delta_k := \mu := 0.001$

shared between different polygons. This implies that corresponding surfaces have common heights $p.z$ in that spot. Then surfaces fit together leading to a watertight roof. But, belonging to different polygons, there also might be vertices with identical x- and y- but different z-coordinates. At such points, surfaces differ in height so that there are step edges (i.e. walls) occurring in the roof.

Because vertices are given with up to three decimal places (millimeters), we propose a concept of approximate planarity of polygon P_k with respect to rounding errors of magnitude $\mu := 0.001$ m. If the polygon lies on a plane with normal vector $\nu = (\nu_k.x, \nu_k.y, \nu_k.z)$, then deviation of coordinates by μ might result in distances up to $\mu(|\nu_k.x| + |\nu_k.y| + |\nu_k.z|)$ between vertices and plane. Our building models have to match footprints from cadastral data. Therefore, we will not modify x- and y-coordinates and have to derive a bound for feasibility of z-coordinates. It will be independent of structural and auxiliary variables. For roof planes, we assume that $\nu_k.z > 0$. Deviation of $p_{k,j}.x$ and $p_{k,j}.y$ by μ leads to a height change of at most $(\sqrt{1-\nu_k.z^2}/\nu_k.z) \cdot \sqrt{2 \cdot \mu^2}$, cf. Fig. 2. We also consider deviation of $p_{k,j}.z$ and call polygon P_k μ-approximate planar if and only if there is a plane such that for all $j \in [m_k]$ the height $p_{k,j}.z$ differs from the z-coordinate of the plane at $(p_{k,j}.x, p_{k,j}.y)$ less than

$$\delta_k := \mu + \frac{\sqrt{1-\nu_k.z^2}}{\nu_k.z} \cdot \sqrt{2} \cdot \mu. \tag{1}$$

This means that each vertex $p_{k,j}$ has to be closer to the plane than $\nu_k.z \cdot \delta_k$.

Current CityGML data sets often violate this definition. The task is to find a function $h : V \to \mathbb{R}$ that maps each vertex to a "better" height, such that polygons with vertices $(p_{k,j}.x, p_{k,j}.y, h(p_{k,j}))$ become μ-approximately planar. Each constant function is such a mapping. However, typical appearances of roofs have to be maintained. This can be achieved by solving the approximation problem to find $h(p) := p.z + h_1(p) - h_2(p)$ such that $h_1(p) \geq 0$ and $h_2(p) \geq 0$ for all $p \in V$, and $\sum_{p \in V} h_1(p) + h_2(p)$ is minimized subject to the linear condition that all surfaces have to be (approximately) planar. To avoid large local errors, we further bound h_1 and h_2 by $h_1(p) \leq \varepsilon$ and $h_2(p) \leq \varepsilon$ using a threshold value $\varepsilon > 0$.

For polygon P_k, we define a plane using three vertices $p_{k,u}$, $p_{k,v}$, and $p_{k,w}$. For numerical stability, we choose indices u and v such that the vertices, if projected to the x-y-plane, have a largest distance. Then $p_{k,w}$ is selected such that the sum of x-y-distances to $p_{k,u}$ and $p_{k,v}$ becomes maximal and vectors $\mathbf{a}_k := (p_{k,u}.x - p_{k,v}.x, p_{k,u}.y - p_{k,v}.y)$ and $\mathbf{b}_k := (p_{k,w}.x - p_{k,v}.x, p_{k,w}.y - p_{k,v}.y)$ become linear independent. We can uniquely write each point $(p_{k,j}.x, p_{k,j}.y)$, $j \in [m_k]$, as $(p_{k,j}.x, p_{k,j}.y) = (p_{k,v}.x, p_{k,v}.y) + r_{k,j}\mathbf{a}_k + s_{k,j}\mathbf{b}_k$, where $r_{k,j}$ and $s_{k,j}$ can be computed, for example, with Cramer's rule. The surface defined by polygon P_k and heights h is planar, if and only if

$$h(p_{k,j}) = h(p_{k,v}) + r_{k,j}(h(p_{k,u}) - h(p_{k,v})) + s_{k,j}(h(p_{k,w}) - h(p_{k,v}))$$

for each $j \in M_k := [m_k] \setminus \{u, v, w\}$. But we only require μ-approximate planarity. Hence, we introduce auxiliary variables $\alpha_{k,j}$,

$$\begin{aligned}\alpha_{k,j} := &-h_1(p_{k,j}) + h_2(p_{k,j}) + (1 - r_{k,j} - s_{k,j})(h_1(p_{k,v}) - h_2(p_{k,v})) \\ &+ r_{k,j}(h_1(p_{k,u}) - h_2(p_{k,u})) + s_{k,j}(h_1(p_{k,w}) - h_2(p_{k,w})) + c_{k,j},\end{aligned} \tag{2}$$

with constants $c_{k,j} := -p_{k,j}.z + (1 - r_{k,j} - s_{k,j})p_{k,v}.z + r_{k,j}p_{k,u}.z + s_{k,j}p_{k,w}.z$ and bounds $-\delta_k \le \alpha_{k,j} \le \delta_k$, see (1).

As mentioned, in V there might be vertices $p_{k,i}$ and $p_{k,j}$ with $p_{k,i}.x = p_{k,j}.x, p_{k,i}.y = p_{k,j}.y$ but $p_{k,i}.z \neq p_{k,j}.z$. At such points, there is a height difference between adjacent roof facets, i.e. there is a step edge that defines a wall between roof segments. For example, such step edges occur with shed roofs or dormers and are very characteristic for the shape of the roof. Therefore, a higher roof segment should still be higher in the outcome of the optimization process. To this end, we determine all sets of vertices with a common pair of x- and y-coordinates. For each set we sort by increasing z-coordinates. Let $v_1, \ldots, v_l \in V$ be the vertices of such a set, so that $v_1.z \le v_2.z \le \cdots \le v_l.z$. Then, for each set, we add following constraints:

$$h(v_2) - h(v_1) \ge 0, \quad h(v_3) - h(v_2) \ge 0, \quad \ldots, \quad h(v_l) - h(v_{l-1}) \ge 0. \tag{3}$$

To summarize, we have constructed a linear program with structural variables $h_1(p), h_2(p)$, $p \in V$, and auxiliary variables $\alpha_{k,j}$, $k \in [n]$, $j \in M_k$:

$$\begin{aligned}\text{Minimize} \quad & \sum_{p \in V} h_1(p) + h_2(p) \\ \text{s.t.} \quad & \text{inequalities (3) hold true, } 0 \le h_1(p), h_2(p) \le \varepsilon \text{ for all } p \in V, \\ & -\delta_k \le \alpha_{k,j} \le \delta_k \text{ for all } k \in [n], j \in M_k, \text{ see (1), (2).}\end{aligned}$$

The use of weights in the objective function has no influence on existence of solutions but determines appearance of the optimized model. Since our research group is interested in facade mapping, changes to vertices of walls should be more expensive than changes to vertices purely belonging to roof facets. Let $w(p) := 2$ if $p \in V$ does

not only belong to a roof facet but also to at least one wall. Otherwise let $w(p) := 1$. Thus, we replace the objective function by

$$\sum_{p \in V} w(p)[h_1(p) + h_2(p)]. \tag{4}$$

Alternatively, one could replace threshold value ε by a function $\varepsilon(p)$, $p \in V$. Additionally, one could consider the number of roof facets, to which a vertex belongs.

Our tool parses CityGML data sets and generates the set V of all roof vertices by searching for CityGML `bldg:RoofSurface` tags that directly correspond to polygons P_k. Then it solves the linear optimization problem using GNU Linear Programming Kit library (GLPK) [9]. If it finds an optimal solution, then it replaces old z-coordinates in the CityGML file with optimized heights, now considering roof and wall vertices.

3 Evaluation

We apply the algorithm to two different CityGML data sets of a German city. We focus on one square kilometer at the heart of the city of Krefeld but also give numbers for the city center's 16 km^2. The files of the square kilometer consist of 4161 buildings. One file is part of the official model driven city model of North Rhine Westphalia [10] and contains 41,496 different LoD2 roof vertices belonging to 4102 buildings. 59 complex buildings are described in LoD1, i.e. with a simplified flat roof. The other data set contains the data driven model [4] with 77,243 vertices and 3987 buildings in LoD2 for the same square kilometer. 174 tiny buildings were not sufficiently covered by laser scanning points and are described in LoD1.

Maximum number of structural variables for a single building is 1388, maximum number of auxiliary variables for one building is 1324. On one kernel of an i5 processor, the running time for the square kilometer is about 5 s.

Results depend on threshold ε. For each building the algorithm starts with $\varepsilon = 0.1$ m. If there is no feasible solution, than optimization is repeated with a doubled ε value. Table 1 and Fig. 3 summarize results. Figure 1 shows the outcome of optimization in connection with objective function (4). To see how numbers scale, we repeat the experiment for 16 km^2, also see Table 1 and Fig. 3. If we iteratively apply the algorithm to its own outcome then we observe a residual number of buildings only becoming planar with $\varepsilon = 0.1$ m. This is due to rounding errors, especially in plane computation, and optimization with float arithmetic. Therefore, column $\varepsilon = 0.1$ m in Table 1 might contain more realistic numbers of approximately planar buildings than column $\varepsilon = 0$ m.

The proposed tool works excellent for a model driven approach with simple standard roofs from a catalogue but it also shows good results for a data driven city model with larger deviations. Our results prove that it is possible to merge vertices during model creation without considering the CityGML planarity requirement and

Table 1 Numbers of approximately planar LoD2 buildings for one and 16 km^2 (UTM intervals [32330000, 32331000] × [5689000, 5690000] and [32329000, 32333000] × [5687000, 5691000])

	ε (m)	0.0	0.1	0.2	0.4	0.8	1.6	3.2	6.4	Rest
1 km^2	Data driven	732	2368	2791	3204	3619	3849	3973	3987	0
	Model driven	3119	4050	4066	4078	4091	4099	4102	4102	0
16 km^2	Data driven	9704	22,446	24,887	27,772	30,183	31,605	32,486	32,575	2
	Model driven	26,721	32,646	32,714	32,772	32,838	32,873	32,884	32,884	0

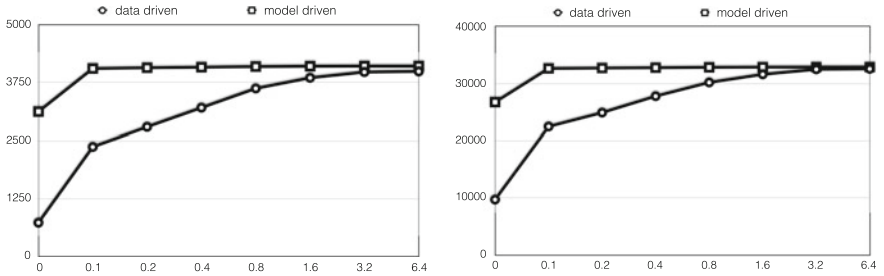


Fig. 3 Numbers of planar buildings for 1 km^2 (*left*) and 16 km^2 (*right*), see Table 1

correct missing planarity later. To get even better results, one could split up perviously merged vertices and introduce additional walls. An integer linear program could select such vertices. Also, the tool could be extended to not only modify height values but x- und y-coordinates, too. However, expected improvement is limited because the building's footprint must not be changed and roof segments have to keep connected.

References

1. Alam, N., Wagner, D., Wewetzer, M., von Falkenhausen, J., Coors, V., Pries, M.: Towards automatic validation and healing of CityGML models for geometric and semantic consistency. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. **II-2/W1**, 1–6 (2013)
2. Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., Maierhofer, S.: O-snap: optimization-based snapping for modeling architecture. ACM Trans. Graph. **32**(1), 6:1–6:15 (2013)
3. Boulch, A., de La Gorce, M., Marlet, R.: Piecewise-planar 3D reconstruction with edge and corner regularization. Comput. Graph. Forum **33**(5), 55–64 (2014)
4. Goebbels, S., Pohle-Fröhlich, R.: Roof reconstruction from airborne laser scanning data based on image processing methods. ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci. **III-3**, 407–414 (2016)
5. Gröger, G., Kolbe, T.H., Nagel, C., Häfele, K.H.: OpenGIS City Geography Markup Language (CityGML) Encoding Standard. Version 2.0.0. Open Geospatial Consortium (2012)

6. He, Y.: Automated 3D Building Modeling from Airborne LIDAR Data. Ph.D. thesis, University of Melbourne, Melbourne (2015)
7. Henn, A., Gröger, G., Stroh, V., Plümer, L.: Model driven reconstruction of roofs from sparse LIDAR point clouds. ISPRS J. Photogramm. Remote Sens. **76**, 17–29 (2013)
8. Kolbe, H.: Representing and exchanging 3D city models with CityGML. In: Lee, J., Ziatanova, S. (eds.) Representing and Exchanging 3D City Models with CityGML. Lecture Notes in Geoinformation and Cartography, pp. 15–31. Springer, Berlin (2009)
9. Makhorin, A.: The GNU Linear Programming Kit (GLPK). Free Software Foundation, Boston, MA (2009)
10. Oestereich, M.: Das 3D-Gebäudemodell im Level of Detail 2 des Landes NRW. Nachrichten aus dem öffentlichen Vermessungswesen Nordrhein-Westfalen **47**(1), 7–13 (2014)
11. Perera, S.N., Maas, N.G.: A topology based approach for the generation and regularization of roof outlines in airborne laser scanning data. DGPF Tagungsband **21**, 1–10 (2012)
12. Wagner, D., Wewetzer, M., Bogdahn, J., Alam, N., Pries, M., Coors, V.: Geometric-semantical consistency validation of CityGML models. In: Pouliot, J., et al. (eds.) Progress and New Trends in 3D Geoinformation Sciences. Lecture Notes in Geoinformation and Cartography, pp. 171–192. Springer, Berlin (2013)

Distributed Solving of Mixed-Integer Programs with GLPK and Thrift

Frank Gurski and Jochen Rethmann

Abstract Branch-and-bound algorithms for Mixed-Integer Programs (MIP) are studied for over 40 years [1, 3, 7]. Object-oriented frameworks for parallel branch-and-bound algorithms like ALPS [9], ParaSCIP [8], and PICO [5] are well known. Our aim is to develop a powerful yet easy-to-use parallel MIP-solver by combining open-source tools or frameworks that are platform independent and free of charge so that even small companies come to the benefit of an optimization suite. Licenses of commercial solvers like CPLEX or GUROBI are often not affordable for small companies. Our tool combines the Gnu Linear Programming Kit (GLPK) and the remote procedure call framework Thrift. To make our development independent of the GLPK-development, we use the GLPK-solvers as independently running processes. So we are able to profit from further development and algorithmic progress of GLPK in future. We describe how to combine these technologies to get an optimization suite for mid-sized problems and evaluate the power of our tool by solving some benchmark data from Chu and Beasley [4] and MIPLIB 2003 [2].

1 Introduction

It is well known that large mixed-integer programs can be solved by sophisticated, massively parallel branch-and-bound based frameworks like ParaSCIP. Such frameworks require much knowledge from the user and adjusting of many parameters is needed in order to solve specific classes of problems efficiently. In contrast to that

F. Gurski
Institute of Computer Science, Heinrich Heine University of Düsseldorf,
40225 Düsseldorf, Germany
e-mail: frank.gurski@hhu.de

J. Rethmann (✉)
Faculty of Electrical Engineering and Computer Science, Niederrhein University
of Applied Sciences, 47805 Krefeld, Germany
e-mail: jochen.rethmann@hs-niederrhein.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_79

we would like to know whether it is possible to build powerful, easy-to-use solvers by using open-source and platform independent software running on ordinary office hardware. Our tool should be easy to use, because small companies often lack special skills in mixed-integer programming. We are interested in the gap between solvers that are easy to use and sophisticated frameworks used in scientific computing. How good and powerful can be a simple solver?

To evaluate the power of our tool we solve benchmark data from MIPLIB 2003 [2] and Chu and Beasley [4]. We compare our tool to the power of CPLEX and we measure parallel scaling performance. The MIP-solver should run in parallel to normal office work without affecting this work. Therefore we have to minimize network traffic and we have to start daemons on idle cpu-cores.

In our explanations we always consider mixed-integer programs of the following form: minimize $c^T x$ subject to $Ax \leq b$ where $x \in \mathbb{Z}^\ell \times \mathbb{R}^{n-\ell}$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$, and $c \in \mathbb{R}^n$. GLPK is able to deal with MIPs given in MPS- and LP-format.

In usual branch-and-bound algorithms an integer variable with fractional value after LP-relaxation is chosen and two new bounds are added to the problem. In contrast to this we choose more than one variable to branch on. We take k variables, compute for each of them two new bounds, build all possible 2^k combinations of these new bounds, and append them to the problem.

To apply the current best upper bound c_{opt} found so far we extend the constraints of the problem by one additional row, namely $c^T x \leq c_{opt}$. Initially, the incumbent value c_{opt} is set to infinity. Adding this constraint to a subproblem possibly makes a subproblem infeasible.

Since Integer Programming is NP-hard [6] there is little hope to find a branching rule and a selection rule that works well for all integer problems. In [7] several branching rules and selection rules are investigated, but none of them clearly dominates the others. A rule that works well for some problems may fail for other problems. Even if we would have had several selection rules, branching rules, or heuristics implemented, there is little hope to choose always the right one for the given problem.

2 Architecture

Our architecture is not designed to allow massively parallel computing. In a typical office of small or mid-sized companies there will be no more than a few tens of computers. Therefore the architecture must not be high scalable, and since load balancing and termination detection are easy, we use a well-known farmer-worker model to parallelize the MIP-solver. Since GLPK is not multi-threaded, we have to start GLPK as an independently running process on each computer. The number of processes should be at most the number of cpu-cores so that the processes really run in parallel. Each worker gets a job from the farmer. Therefore the farmer has a pool of jobs. The communication between farmer and worker is done by remote procedure

```
glp_prob *lp = glp_read_file(...);
int rc = glp_simplex(lp);
if (rc != 0) report failure and stop;
pool = createNewJobs(lp);
for (i = 0; i < numWorker; i++)
    thread t(proxy, i);
```

Fig. 1 Pseudo-C-code fragment of farmer

```
worker.setInstance(lp);
while (!stopped and !pool.empty()) {
    job = pool.pop();         // must be thread-safe
    rc = worker.solve(job, curOptVal);
    if (rc.err == 0 or rc.status == FEASIBLE)
        updateValue(rc.value);       // thread-safe
    if (rc.err == TIMEOUT)
        for each job in rc.joblist do
            pool.push(job);  // must be thread-safe
}
```

Fig. 2 Pseudo-C-code fragment of worker-proxy

calls via the Thrift framework which was originally developed by Facebook, and is nowadays available under an Apache licence.

Farmer First the farmer reads the problem from file. Afterwards it solves the LP-relaxation of the problem by the method `glp_simplex`. This step is necessary to determine integer variables that have fractional value. Some of these variables are chosen and the farmer creates jobs by adding bounds over the chosen variables. A pseudo-C-code fragment of the farmer is shown in Fig. 1. Finally threads are started to communicate with the workers. These threads are called worker-proxies. Initially, the farmer produces always at least as much jobs as workers are available.

Worker-proxy As long as there are jobs in the pool each worker-proxy selects one subproblem and sends it to its associated worker. The pool of jobs is realized as a priority queue. A pseudo-C-code fragment of the worker-proxy is given in Fig. 2. The worker-proxy first calls the function `setInstance` on the worker. The problem is sent only once to the worker to reduce network traffic. After that only the jobs, i.e. the additional bounds on the variables, are sent to the worker. If the worker finds a solution, this solution is sent back to the worker-proxy. At the proxy it is checked whether this solution is better than the best solution c_{opt} found so far and it is stored in the variable `curOptVal` when required, which is done by the method `updateValue`.

```
glp_set_row_bnds(addedRow, curOptVal);
rc = glp_simplex(job);
if (rc == 0 and glp_status() == FEASIBLE) {
    res.err = glp_intopt(job);
    res.status = glp_status();
    if (res.err == 0 or res.status == FEASIBLE)
        res.value = glp_get_obj_val(job);
    if (res.err == TIMEOUT)
        res.joblist = createNewJobs(job);
} else res.err = CUT_BY_SIMPLEX;
return res;
```

Fig. 3 Pseudo-C-code fragment of the method `solve(job, curOptVal)` at worker

In GLPK we have the ability to restrict the running time of the MIP-solver at starting time. If the MIP-solver needs more time, it is stopped automatically. In this case the worker generates some new jobs by adding bounds to the old job, sends these new jobs back to the worker-proxy, and the worker-proxy puts them into the job-pool. If the MIP-solver has found a feasible solution this is also sent back to the worker-proxy and the value of this solution has to be stored in variable `curOptVal` at the farmer as a new incumbent value when required.

Worker The function `setInstance` on the worker stores the problem to be solved for future calls of function `solve`, it does the pre-processing like computing a branching order, and it adds the row $c^T x \leq c_{opt}$ to the constraints of the original problem by calling the method `glp_add_rows`.

To solve a subproblem the worker first puts the incumbent c_{opt} as a bound to the additional row, see Fig. 3 for a pseudo-C-code. Then it solves the LP-relaxation by calling the `glp_simplex` method. If no feasible solution has been found, or in other words if the dual bound is worse than the best solution c_{opt} found so far then there is no need to call the MIP-solver. Otherwise, the MIP-solver `glp_intopt` is called. If the time of the MIP-solver is expired then the best feasible solution is returned and new jobs must be created and returned. If a solution of the subproblem was found in time, it is sent back to the farmer, otherwise the farmer is informed about the failure.

Subproblem selection rules While the job-pool is not empty a worker-proxy has to select a job from the pool and send it to its associated worker. In the literature many job selection rules like Breadth-First Search (BFS), Depth-First Search (DFS), Best-Bound Search (BBS), and even more complex rules like estimate-based methods are studied. DFS performs poorly in practice [7]. BFS often takes too much space, since many of the primary generated jobs cannot run to completion and therefore many jobs are produced. So we decided to choose BBS.

To compute bounds of the subproblems for BBS we use the simplex method of GLPK. This may be time consuming but this selection rule reduces the number of subproblems in the job-pool so that the memory consumption hopefully is small and so is the influence on office work.

Branching rules In the literature many branching rules like pseudo-cost, full strong and most fractional branching are studied. In [1] it is shown that most fractional branching is in general not better than selecting a variable randomly. Our rule comes from [3]: The integer variables are processed in the decreasing order of their absolute cost values in the objective function. This is computed on each worker in a pre-processing phase only once.

3 Experimental Results

The farmer is always started on a dedicated computer. Since the farmer has to store many jobs in its pool it should have large physical memory, at least 16 GB. The MIP-solver uses presolving and Gomory cuts, it is running on common office hardware using Ubuntu Linux 14.04 (LTS), kernel 3.16, GLPK 4.52, Gnu C++-compiler 4.8.4, and Thrift 0.9.3. Two parameters of our tool must be set: the time-limit of the MIP-solver and the number of jobs to generate in case of a worker time-out. After some measurements it turns out to be a good compromise to choose 10 s as the time-limit for the MIP-solver and to generate 4 times more jobs as workers if the MIP-solver has not run to completion. To prevent a memory overflow the number of generated jobs is reduced if more than half of the total memory on farmer is used. To decrease network traffic the farmer removes all jobs from the pool whose dual bound is worse than the incumbent value c_{opt}. This is done in function `updateValue`.

Parallel scaling performance In Table 1 the speedup is shown for instances of Chu and Beasley's benchmark set and for instances of MIPLIB 2003. We have used up to 16×4 workers, the comparison with CPLEX version 12.6.0.0 using default settings running on a common office computer with 8 GB RAM, Debian Linux 6.0.10, and 4 threads is given too.

4 Conclusion and Acknowledgment

In our future work we will implement checkpointing and restarting. This is common in parallel computing to protect against failures.

Our algorithm performs non-deterministically: Two executions might follow different paths in the search tree, produce different solutions, and need different solving times. Practitioners do not accept such non-determinism, so we will implement deterministic solving, and we will study the resulting performance degradation.

We would like to thank Jochen Peters for some first implementations of our tool during the work on his bachelor thesis, and Jens Gräbel for some fruitful discussions.

Table 1 Parallel scaling performance. Running times are given in seconds, times greater than 2 h are indicated by a bar. Sequential running time of GLPK is always greater than 3 h, except pk1 is solved in 5056 s and qiu in 2309 s. In the last row we count the instances that our parallel solver computes at least as fast as CPLEX

Instance	4×4	8×4	12×4	16×4	CPLEX
OR30x100-0.50_2	1041	498	322	244	456
OR30x100-0.50_3	1341	682	406	311	626
OR30x100-0.25_9	2111	1426	920	703	1098
OR10x250-0.75_1	3000	1459	982	750	755
OR10x250-0.75_2	–	4177	3243	2434	3160
OR10x250-0.75_3	2380	1278	826	624	478
OR10x250-0.75_4	1421	741	446	338	413
OR10x250-0.75_5	3597	1627	1032	782	1198
OR10x250-0.75_6	1844	920	609	461	381
OR10x250-0.75_9	1362	701	466	353	276
OR10x250-0.25_2	4875	1449	982	744	3003
OR10x250-0.25_3	4091	2081	1445	1099	1224
OR10x250-0.25_6	5341	2733	1964	1488	1568
OR10x250-0.25_7	4544	2427	1676	1273	1028
OR10x250-0.25_9	–	5519	4339	3236	3603
danoint	601	328	222	171	780
mas74	4172	2016	1095	830	224
mas76	116	50	37	31	14
modglob	1677	1041	823	630	1
pk1	66	30	22	20	17
qiu	258	132	109	90	4
As fast as CPLEX	1	2	6	12	

References

1. Achterberg, T., Koch, T., Martin, A.: Branching rules revisited. Oper. Res. Lett. **33**, 42–54 (2005)
2. Achterberg, T., Koch, T., Martin, A.: MIPLIB 2003. Oper. Res. Lett. **34**(4), 361–372 (2006)
3. Benichou, M., Gauthier, J., Girodet, P., Hentges, G., Ribiere, G., Vincent, O.: Experiments in mixed-integer linear programming. Math. Program. **1**, 76–94 (1971)
4. Chu, P., Beasley, J.: A genetic algorithm for the multidimensional knapsack problem. J. Heurist. **4**(1), 63–86 (1998)
5. Eckstein, J., Phillips, C.A., Hart, W.E.: Pico: An object-oriented framework for parallel branch and bound. In: Dan Butnariu, Y.C., Reich, S. (eds.) Inherently Parallel Algorithms in Feasibility and Optimization and their Applications, Studies in Computational Mathematics, vol. 8, pp. 219–265. Elsevier (2001)
6. Garey, M., Johnson, D.: Comput. Intractability. W.H. Freeman and Company, San Francisco (1979)
7. Linderoth, J., Savelsbergh, M.: A computational study of search strategies for mixed integer programming. INFORMS J. Comput. **11**(2), 173–187 (1999)

8. Shinano, Y., Achterberg, T., Berthold, T., Heinz, S., Koch, T.: ParaSCIP: A parallel extension of SCIP. In: Bischof, C., Hegering, H.G., Nagel, E.W., Wittum, G. (eds.) Competence in High Performance Computing, June 2010, pp. 135–148. Springer, Berlin (2012)
9. Xu, Y., Ralphs, T.K., Ladányi, L., Saltzman, M.J.: Alps: A framework for implementing parallel tree search algorithms. In: Golden, B., Raghavan, S., Wasil, E. (eds.) The Next Wave in Computing, Optimization, and Decision Technologies, pp. 319–334. Springer, US (2005)

Extension of Mittelmann's Benchmarks: Comparing the Solvers of SAS and Gurobi

Werner E. Helm and Jan-Erik Justkowiak

Abstract SAS Institute has long been recognized by Gartner (most recently in 2016, see Kart et al. Magic Quadrant for Advanced Analytics Platforms (09 February 2016|ID:G00275788), [1]) as leader in the magic quadrant for Advanced Analytics Platforms. Bundled in SAS/OR are the Institute's offerings for solving a broad range of OR-problems on various software platforms. We present what appears to be the first independent benchmarking of SAS Institute's OR-solvers. In the present paper we concentrate on the problem classes LP, MILP, Network and Infeasibility Detection.

1 Introduction

As each and every software vendor tries to present its own products in a best possible way truly independent comparisons (benchmarks) are an invaluable source of information for any decision maker who decides upon a company's IT. With regard to OR-Software and solvers for LP-, MILP-, Network- and related problems Hans D. Mittelmann of Arizona State University has almost gained cult status: several times a year he publishes benchmarks that define the state-of-the-art. Mittelmann's benchmarking results (see [2]) in turn are being used as marketing instruments by vendors. During the years 2010–2015 his benchmarks documented the way of a then new company (albeit being based on decades of prior experience) Gurobi to the absolute top of performance in most if not all (sub-)categories of common problems. There is no easy and direct way to assess or compare market shares (volume or dollars). Some company people speak of the 'big four' of commercial vendors comprising (could be disputed) IBM ILOG - CPLEX, FICO - XPRESS, Gurobi - Gurobi Solvers and SAS Institute - SAS/OR. Up to most recently only the first three (plus MOSEK,

W.E. Helm (✉)
Stat + OR, FB MN, Hochschule Darmstadt, Darmstadt, Germany
e-mail: Werner.Helm@h-da.de

J.-E. Justkowiak
FB MN, Hochschule Darmstadt, Darmstadt, Germany
e-mail: jan-erik.justkowiak@stud.h-da.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_80

Matlab and open source products) had been covered by H.D. Mittelmann. Hence we set out to extend this range to cover SAS/OR, too. As benchmarks depend on a large number of parameter settings starting from hardware, OS, etc. we directly compared SAS/OR with Gurobi and used our results on Gurobi to link these numbers to Mittelmann's suite of benchmarks. Derived from recent projects we added a class of vehicle routing problems (VRP); results will be published elsewhere.

2 Benchmarks: Basic Setup

As we want to generate results for solvers in SAS/OR that are comparable to Hans D. Mittelmann, we use the same basic setup, which we summarize very briefly (all details are on the web pages [2]). Benchmarks are done in categories (classes) and subcategories (subclasses), e.g. MILPs (mixed integer linear programs) and Infeasibility Detection for MILPs. Basic performance measure is run time in the form of elapsed time or wall-clock time including the input of the problem data from mps-files. In the class of MILPs a solution checker is used by Mittelmann [2]. After all run times are collected the key performance measure of a solver for a specific problem class P is the following shifted geometric mean with shift factor r. Let t_p^s denote the run time to solve problem p by solver s (from solver collection S). The shifted geometric mean for solver s to which we simply refer as geo-mean is then defined by

$$\bar{t}_s := \sqrt[n]{\prod_{p \in P} (t_p^s + r)} - r,$$

where n is the number of problem instances in problem class P. Of course $\bar{t}_s = \bar{t}_s(P)$, i.e. the value depends on the problem class P under consideration. We use a shifting of $r = 10$ s. As in Mittelmann we then apply the following scaling $\bar{t}_S^* = \min_{s \in S} \bar{t}_s$, to calculate for each solver its scaled shifted geometric mean $\tilde{t}_s = \frac{\bar{t}_s}{\bar{t}_S^*}$, which we denote in tables below simply by scaling. This value shows the factor that a specific solver is slower than the best one with scaled value of 1.00. Each problem class is given a maximal time value; a solver's performance on a particular instance is graded success if it solves the instance to optimality within that maximal time. Otherwise success is denied and the maximal value is used. The maximal values being used are LP (25,000 s), MILP (7200 s), Network (3600 s), Infeasibility Detection (3600 s). Concerning the choice of Simplex variant we respected the systems' default settings (Dual Simplex). As SAS historically has used resources-saving defaults we had to activate the memsize-option to enable the full usage of available RAM. In the following we concentrate on the classes LP, MILP, Network and Infeasibility Detection. Most problem instances and reference solutions are taken from MIPLIB [3] or directly from Mittelmann [2].

Table 1 Direct comparison SAS/OR and Gurobi

Summary of results	Success rate		Geo-mean[a]		Scaling	
Problem class	SAS	Gurobi	SAS	Gurobi	SAS	Gurobi
LP	98%	100%	163.82	51.29	3.20	1.00
MILP (1 thread)	85%	99%	432.50	78.81	5.49	1.00
MILP (4 threads)	87%	100%	226.91	45.88	4.95	1.00
MILP (12 threads)	89%	99%	240.27	53.57	4.49	1.00
NETWORK[b]	100%	100%	48.39	17.08	2.83	1.00
NETWORK[c]	100%	100%	14.52	17.08	1.00	1.18
INFEASIBILITY	78%	100%	297.72	17.01	17.50	1.00
VRP	73%	80%	542.37	272.30	1.99	1.00

[a]Shifted geometric mean of run times (elapsed time)
[b]SAS Proc Optlp - Dual Simplex
[c]SAS Proc Optlp - Network Simplex

2.1 Direct Comparison SAS/OR - Gurobi

All direct comparisons were done on a PC with Intel i7-4790 cpu @3.60 GHz (4 cores) with 24 GB RAM under Windows 7 X64 between SAS 9.4 TS1M3 - Analytics 14.1 and Gurobi 6.5.0. We used Python scripts to drive Gurobi [4]. Out of SAS/OR [5] the procedures OPTLP, OPTMILP and OPTNET were applied. Concerning SAS several peculiarities had to be mastered. We mention just one: SAS does not natively read mps-files but circulates a macro to convert external mps-files into an internal mps-format, a process that frequently requires special attention. Hence all mps-files were first converted and manually checked and only then read in to the solver from this internal mps-format.

Table 1 contains the main results of the direct comparison. With the exception of network problems Gurobi outperforms SAS by a considerable margin (scaling factors of around 3 for LP-, around 5 for MILP-problems and around 17 for infeasibility detection). The positive exception for SAS are network problems due to the Network Simplex being implemented in the dedicated procedure Proc Optlp. Figures 1 and 2 display a more detailed comparison showing the robustness of the shifted geometric mean when compared to the arithmetic mean (denoted by a star).

3 Linking the Results to Mittelmann's

It is well known that a uniform behaviour of all solvers across different machines, different amounts of memory, different hard disks and operating systems and different problems cannot be expected. Most of Mittelmann's benchmarks are being run on a PC with Intel i7-4790K (4 cores) @4.0 GHz with 32 GB RAM under Linux. As we ran the same version of Gurobi on our slightly less powerful system under

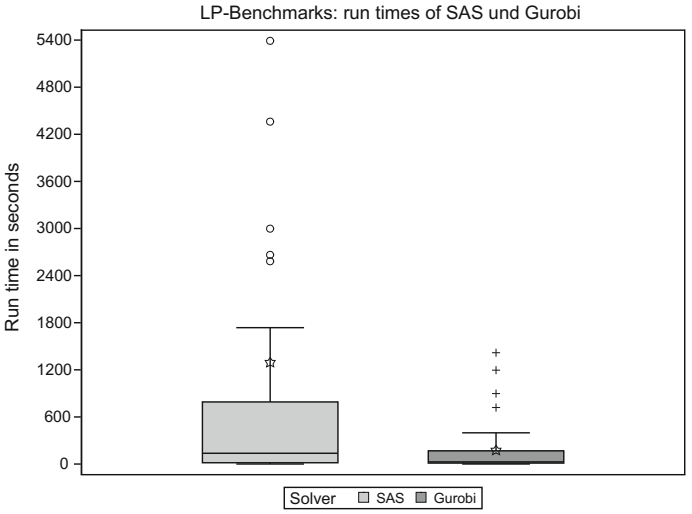


Fig. 1 Boxplot of run times for 40 LP instances; one data point for SAS with value 25,000 (problem not solved) is not visible on plot, but included in the evaluation

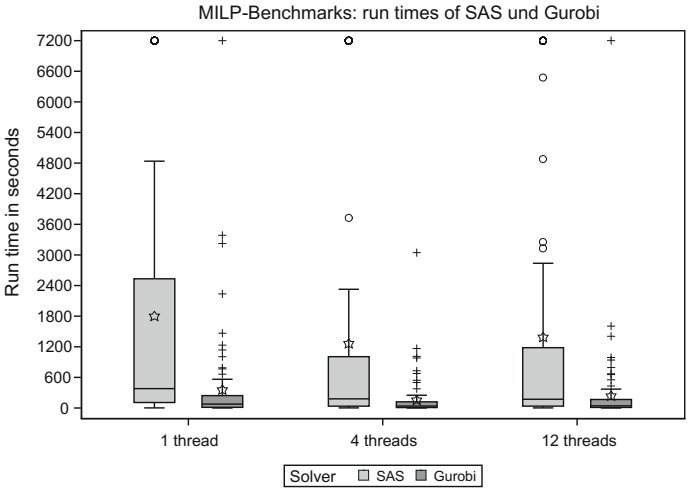


Fig. 2 Boxplot of run times for 87 MILP instances. 12 threads are beyond the hardware's true capabilities, hence performance is degrading. Frequently the best performance was observed when we let the number of threads chosen within the software equal the number of cores of the Intel chip

Windows 7 for each problem class we computed a conversion factor (in the range 0.73–0.85) to upscale our results and make them comparable to Mittelmann. Hence we add a column of adjusted SAS-values to Mittelmann's tables and discuss the results.

Table 2 Performance - LP

SOLVER	CPX	GRB	MSK	XPS	CLP	LPSLV	GLPK	MTB	SAS[a]
Success (of 40)	38	40	40	40	40	23	28	31	39
Geo-mean[b]	82.90	42.80	106.00	41.50	42.90	5068.00	1834.00	485.00	135.97
Scaling	2.00	1.03	2.55	1.00	1.04	122.00	44.20	11.70	3.28

[a]Adjusted by conversion factor of 0.83
[b]Shifted geometric mean

Table 3 Performance - MILP (4 threads)

SOLVER	CBC	CPX	FSCIPC	FSCIPS	GRB	XPS	SAS[a]
Success (of 87)	61	86	75	69	87	87	76
Geo-mean[b]	1053.00	46.00	339.00	641.00	39.00	48.00	192.87
Scaling	27.40	1.19	8.82	17.00	1.00	1.25	4.95

[a]Adjusted by conversion factor of 0.85
[b]Shifted geometric mean

We use the abbreviations GRB (Gurobi), CPX (CPLEX), XPS (Xpress), CLP and CBC (Coin-OR), MSK (MOSEK), SCIPC, SCIPS, FSCIPC and FSCIPS (SCIP family), GLPK (GNU LP Kit), LPSLV (LP Solve), MTB (Matlab), QSOPT (U Waterloo), SAS (SAS/OR) and the colloquial phrase 'Big Three' for the set GRB, CPX, XPS. Further details are found on [2]. When linking our results to Mittelmann's [2] we took his benchmarks as of April, 6, 2016 with the following publication dates LP (April, 2nd, 2016), MILP (April, 4, 2016), Network (November, 25, 2015), Infeasibility Detection (February, 29, 2016). Table 2 displays the results of the 40 LP-problems. Xpress, Gurobi and remarkably CLP (Coin-OR) are the clear leaders. SAS comes in behind MOSEK. Table 3 (and tables 3A, 3B (omitted)) contain results for 87 MILPs differing in the number of threads being activated. The commercial Big Three are in the lead with Gurobi in the top position. With some margin SAS ranks fourth, slightly before SCIPC, but markedly before the rest of the field. Going from 1 to 4 threads improves performance by a factor of two with no effect on the ranking. The results of Mittelmann show that further performance gains can be realized when using more threads on appropriate hardware (more cores). He utilizes an Intel Xeon X5680 (12 Threads), 3.33 GHz Processor instead of his main PC. However, it is unclear if the solvers of all competitors scale in the same way and when to expect a change in the ranking of the top 4 or 5 contenders. There are very promising attempts to move on supercomputers into the range of thousands to a million cores (see [6, 7] who solved 12 previously unsolved MIP-instances from MIPLIB with an enhanced version of ParaSCIP, using up to 80,000 cores).

Table 4 displays results for 10 network problems in default mode (Dual Simplex). The results are similar to those for LP problems with SAS ranking behind the Big Three as well as behind MOSEK and CLP. Using the Network Simplex SAS Proc Optlp achieves the best geo-mean of all tested solvers (see Table 5). The commercial solvers are close together. Results for 18 infeasible problems show superior performance of the Big Three over all other solvers including those of SAS.

Table 4 Performance - Network (Proc Optlp - Dual Simplex)

SOLVER	CPX	MSK	CLP	QSOPT	GRB	XPS	SAS[a]
Success (of 10)	10	10	10	6	10	10	10
Geo-mean[b]	13.22	17.73	29.03	416.43	12.42	25.37	35.32
Scaling	1.06	1.43	2.34	33.50	1.00	2.04	2.83

[a]Adjusted by conversion factor of 0.73
[b]Shifted geometric mean

Table 5 Performance - Network (Proc Optlp - Network Simplex)

SOLVER	CPX	MSK	CLP	QSOPT	GRB	XPS	SAS[a]
Success (of 10)	10	10	10	6	10	10	10
Geo-mean[b]	13.22	17.73	29.03	416.43	12.42	25.37	10.60
Scaling	1.25	1.67	2.74	39.29	1.18	2.29	1.00

[a]Adjusted by conversion factor of 0.73
[b]Shifted geometric mean

4 Conclusion and Outlook

Based on subjective impressions the SAS/OR solvers have improved in recent years in scope and in performance. Due to the lack of benchmarks until now one cannot say if the distance to the Big Three and to Gurobi in particular is increasing or decreasing. It appears that the developers at SAS do not only focus on speed but on a broad leverage of OR-technology into different parts of the SAS system. SAS Institute has provided information that considerable improvements can be expected with its Analytics 14.2 release (due in late 2016). We are working on a solution checker for the SAS benchmarks and plan to publish results for new SAS/OR releases. Additionally we will try to cover the High-Performance Optimization of SAS running in distributed mode.

References

1. Kart, L., Herschel, G., Linden, A., Hare, J.: Magic Quadrant for Advanced Analytics Platforms (09 February 2016|ID:G00275788). http://www.gartner.com/technology/topics/data-analytics.jsp (2016)
2. Mittelmann. H.: Benchmarks for Optimization Software. http://plato.asu.edu/bench.html (2016)
3. Koch, Th, Achterberg, T., et al.: MIPLIB 2010—mixed integer programming library version 5. Math. Progr. Comput. **3**, 103–163 (2011)
4. Gurobi Optimization Inc.: Gurobi Optimizer Reference Manual. http://www.gurobi.com/documentation/ (2015)
5. SAS Institute Inc.: SAS/OR 14.1 Documentation. https://support.sas.com/documentation/onlinedoc/or/ (2016)

6. Koch, Th, Ralphs, T., Shinano, Y.: Could we use a million cores to solve an integer program? Math. Methods Oper. Res. **76**, 67–93 (2012)
7. Shinano, Y., Achterberg, T., Berthold, T., Heinz, S., Koch, Th., Winkler, M.: Solving Open MIP Instances with ParaSCIP on Supercomputers using up to 80,000 Cores. International Parallel and Distributed Processing Symposium (IPDPS), Chicago, IL, pp. 770–779 (2016)

Part XX
Supply Chain Management

3D Printing as an Alternative Supply Option in Spare Parts Inventory Management

Marko Jakšič and Peter Trkman

Abstract We study a spare parts stochastic production/inventory problem of a manufacturer, where he has an option to source parts from a relatively inflexible conventional regular supplier ordering in large batches with long replenishment lead time, or alternatively from a flexible in-house or outsourced 3D printing facility. We derive the dynamic programming formulation for cost associated with utilizing the 3D printing supply option and give some insights into the structure of the optimal policy. In a numerical experiment, we compare the performance of this hybrid sourcing strategy with the regular sourcing option, and provide some managerial insights.

1 Introduction

3D printing, also commonly referred to as additive manufacturing, has lately received considerable attention in practice and research community. 3D printing enables small quantities of customized goods to be produced at relatively low costs and has been compared to such disruptive technologies as digital books and music downloads that enable consumers to order their selections online, allow firms to profitably serve small market segments, and enable companies to operate with little or no unsold finished goods inventory [1]. However, the way that the quickly developing technology will earn its spot in production environments and supply chains usually revolves around a radical redesign of conventional manufacturing and supply chain processes. We take a different approach, and explore how novel technology can be combined with existing operational practices to attain imminent operational benefits.

Our research is motivated by the spare parts inventory control problem faced by one of the leading European home appliance manufacturers. The company has recently undergone a major redesign of their spare parts distribution system by moving from a decentralized distribution system to a central spare parts storage location, while at the same time offering a direct replenishment of spare parts to the

M. Jakšič (✉) · P. Trkman
Faculty of Economics, University of Ljubljana,
Kardeljeva Ploščad 17, Ljubljana, Slovenia
e-mail: marko.jaksic@ef.uni-lj.si

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_81

field-technicians. While this has led to considerable savings particularly related to spare parts inventory holding costs, it has also provided the company the opportunity to centralize their spare parts production capacities. The company currently still strongly relies on off-shore production facilities for replenishment of spare parts, where for plastic parts the injection moulding is the prevalent technology.

Inline with the goal of continually reducing the inventory levels of spare parts, while maintaining near perfect service level to their customers, the company is exploring the opportunity to manufacture parts on demand by using the 3D printing technology. However, while the move towards 3D printing has not posed large technological difficulties for a relatively large assortment of spare parts, the major obstacles are relatively more expensive per part production costs and the lack of 3D printing capacity availability. In order to keep the paper concise, we direct the reader to [2–5] for the description of the fundamental differences in both manufacturing approaches.

Several papers discuss the impact of additive manufacturing in spare parts supply chains [2, 6–8]. These papers focus on the trade-offs involved in switching to a new manufacturing technology, where on demand and centralized production is proposed as the most likely scenario currently. Until 3D printing becomes a more general purpose technology, a full switch to 3D printing is not expected in most applications. In spare parts supply in particular, the availability of mass production resources even after the regular production has stopped generally ensures a relatively cheap supply of spare parts. Therefore, the objective of this paper is to study the possibility of using regular supply and possibly capacity constrained 3D printing supply option simultaneously to ensure adequate supply of spare parts.

There has not been notable effort made by the inventory research community to study 3D printing as an alternative supply option, and its effect on the inventory control policy on a operational level. Apart from [9], authors rely to simple safety stock calculations to assess the potential benefits of 3D printing [7, 8]. This can be attributed to the fact that 3D printing, as an alternative supply option, fits well within the two closely related groups of inventory models: emergency replenishment models [10–13], and dual sourcing inventory models [14–17]. Most of the papers we refer to also incorporate the capacity constraint on the alternative supply option, which is particularly relevant for our paper.

2 Model Formulation and Inventory Policy Characterization

In this section, we give the notation, the model description and provide some insights into the properties of the proposed inventory policy. We model a spare parts inventory system where a stationary stochastic demand for spare parts is assumed. We denote the demand realization in period t as d_t and the distribution function of the demand as $D(\cdot)$. The inventories are normally replenished from the regular produc-

tion at the cost of c_r per part. As the regular supply channel using injection molding is characterized by large production batches, it is reasonable to assume that the need to replenish spare parts from an alternative supply source possibly only emerges towards the end of the regular replenishment cycle, shortly before the regular production order is delivered.

We focus on modeling the alternative 3D supply option, where the parts can be produced internally with the company's own 3D printing capabilities at the cost c_p per unit, or sourced from an outsourced 3D printing facility at a cost c_e per unit. As the costs of outsourced production are assumed to be higher, $c_r < c_p < c_e$, this supply option is considered as an emergency 3D printing option, while the in-house production is considered as a primary 3D printing supply source. Given the 100% service level guarantee, we assume that the combined 3D printing production capacity is always sufficient to cover the current period's demand. If in-house production capacity is not sufficient (or completely unavailable) on a particular day, the fully available outsourcing supply option is utilized to meet the demand. We assume stationary stochastic in-house production capacity, where the capacity availability in period t is denoted as q_t, and the distribution function of capacity as $Q(\cdot)$.

Presuming that demand is always met with certainty, the goal is to find a 3D printing replenishment policy that would minimize the production/sourcing costs, and the inventory holding costs in the periods until the end of the regular replenishment cycle. In a period t, prebuilding inventories might be desired to cope with future periods of possibly insufficient in-house capacity, instead of relying to a more expensive outsourcing supply option. This results in a positive starting inventory position x_{t+1} in the following period.

Rather than waiting till the inventories of spare parts from regular replenishment are depleted to zero, the policy should give the inventory level at which it is optimal to start utilizing the 3D printing supply option. In addition, the policy is determined by the optimal starting inventory position $\hat{x}_t$ for each of the subsequent periods before the arrival of the next regular order. We denote the period in which the 3D printing supply option is used first as period k, where k represents the number of periods till the end of the regular replenishment cycle ($k = 0$). Our objective is to determine the optimal starting inventory positions $(\hat{x}_k, \hat{x}_{k-1}, \ldots, \hat{x}_2, \hat{x}_1)$. It is expected that the need to prebuild the inventories diminishes as k decreases towards 0 (where for $k = 0$, $\hat{x}_0 = 0$ is optimal), therefore intuitively the optimal starting inventory positions should be ordered in the following way: $\hat{x}_k \geq \hat{x}_{k-1} \geq \cdots \geq \hat{x}_2 \geq \hat{x}_1$.

We assume the following sequence of events: (1) At the start of period t, the demand d_t for spare parts that need to be replenished in the current period is revealed. Current inventory position x_t and the available in-house 3D printing production capacity q_t are checked. (2) Depending on the replenishment requirements, the order z_t is placed at most up to q_t in the case where $z_t \leq q_t$, while in the case of $z_t > q_t$ the outsourcing option is used to cover the excess part of the order $(z_t - q_t)^+$. (3) The order z_t from in-house production, and possibly outsourcing supply option, is replenished, and the production costs are charged. The demand is satisfied and the inventory holding costs are charged based on the end-of-period inventory position, $h(x_t + z_t - d_t)^+$, where h denotes the per unit per period inventory holding costs.

The single period cost function $C_t(x_t, z_t, q_t, d_t)$ can be written as:

$$C_t(x_t, z_t, q_t, d_t) = c_p \min(z_t, q_t) + c_e(z_t - q_t)^+ + h(x_t + z_t - d_t)^+, \tag{1}$$

where the first term denotes the in-house production costs charged in the production quantity that is potentially limited by the available capacity, the second term denotes the outsourcing costs associated with the part of the order in excess of in-house capacity, and the third term denotes the inventory holding costs charged on the end-of-period inventory position. Recall that the end-of-period inventory position (that also corresponds to the starting inventory position x_{t+1} in period $t + 1$) cannot be negative due to the 100% service level assumption.

As available capacity q_t and demand d_t are known prior to making the ordering decision, the policy is straightforward for the single period problem. It instructs that the demand is satisfied by utilizing the in-house production capacity first, and it resorts to the outsourcer only if in-house capacity is insufficient. Also, it is not rational to source spare parts in the quantity that exceeds the demand as this leads to inventory holding costs. The decision maker would follow such policy in the last period before the arrival of the regular order.

As already pointed out above, prebuilding of inventories is a viable option in a multiperiod setting. Producing above the demand d_t in period t results in increased starting inventory position x_{t+1} in the following period. This enables the system to cope better with the possible capacity shortages in the following periods, and avoid using the expensive outsourcing supply option. Thus, intuitively, outsourcing supply option is never used to prebuild inventories, but it is utilized only in the case when in-house is insufficient to cover current period's demand.

Correspondingly, the minimal discounted expected cost function associated with using a 3D printing supply option that optimizes the cost over a finite planning horizon T from period t onward, starting in the initial state x_t, can be written as:

$$f_t(x_t) = E_{Q_t, D_t} \tilde{f}_t(x_t, q_t, d_t) = \min_{x_t + z_t \geq d_t} \left\{ C_t(x_t, z_t, q_t, d_t) + \alpha f_{t+1}(x_{t+1}) \right\}, \tag{2}$$

and the ending condition is defined as $f_{T+1}(x_{T+1}) = h x_{T+1}^+$. Period $T + 1$ denotes the period in which the regular replenishment arrives ($k = 0$), where the inventory holding are charged in the likelihood that any inventory is carried over.

3 Numerical Study

In this section, we present the results of the numerical analysis, which we carried out to determine the optimal starting inventory positions $\hat{x}_k$, and the benefits of the hybrid 3D sourcing option. Numerical calculations were done by solving the dynamic programming formulation given in (2) to determine the performance within a regular replenishment cycle.

Table 1 (a) The optimal safety stock levels, and (b) the cost benefits of 3D sourcing

CV_D	Sourcing type	Optimal safety stock level									Cost savings		
		k	30	25	20	15	10	5	1	p	0.1 (%)	0.5 (%)	0.9 (%)
0.5	Regular		13	12	12	11	10	8	5		1.53	1.54	2.25
	Hybrid		11	10	10	9	8	7	5				
1	Regular		27	26	25	23	21	18	13		2.77	3.44	3.46
	Hybrid		22	22	21	20	18	16	12				
2	Regular		52	51	49	47	44	40	32		5.28	5.87	6.46
	Hybrid		42	41	41	39	38	35	30				

We present the results in Table 1, where we use the following set of parameters. The demand for spare parts was modeled using Gamma distribution with expected demand of 2 units per day, and the coefficients of variation $CV_D = \{0, 0.5, 1, 2\}$. The primary in-house printing capacity is of "all-or-nothing" type Bernoulli distribution, with probability of fully available capacity on a particular day $p = \{0.1, 0.5, 0.9\}$. We assume the following production costs: $c_m = 5$, $c_p = 8$ and $c_e = 10$; and the holding cost $h = 0.008$ per day. The regular replenishment is characterized by the order quantity $Q = 125$, which corresponds to 4 replenishment cycles per year, and the lead time of 30 days is assumed.

As expected, we see in Table 1(a) that the optimal safety stock levels, defined as $\hat{x}_k - \sum_{j=1}^{k} E(D_j)$, are higher for the regular single sourcing supply option. The hybrid sourcing option, which is (in addition to regular supply) using both 3D printing options, saves on inventory holding costs at the expense of relatively higher production costs of utilizing 3D printing when needed. The savings of the hybrid policy are positive despite the relatively large difference in per part production costs, and are higher in the case of higher demand variability and higher availability of the primary in-house 3D printing capacity.

4 Conclusions

In this paper we derive the model formulation and characterize the optimal inventory policy for the inventory model with two supply sources. The regular supply source is typical inflexible large batch production with long lead time, while in the case of imminent stockout, the alternative 3D printing supply option is utilized. 3D printing technology enables responsive production, either in-house or at the outsourced production facility. Here we point out that the optimal policy does not have a simple order-up-to structure, which is attributed to two major system characteristics: the similarity to the lost sales inventory models and the additional complexity due to the assumption of stochastic production capacity constraint.

As we only provide limited insights within this short paper, the research work is generally targeted towards an analytical characterization of the target inventory levels. In addition, based on our observations from practice, there is a need for a more detailed modeling of the production capacity associated with 3D printing. In this paper, we assume that the production capacity availability is exogenous to system, while in reality it is a direct consequence of the capacity level decisions on a strategic level, and the capacity allocation decisions on a daily operational level.

References

1. Berman, B.: 3-d printing: the new industrial revolution. Bus. Horiz. **55**(2), 155–162 (2012)
2. Pérès, F., Noyes, D.: Envisioning e-logistics developments: making spare parts in situ and on demand: state of the art and guidelines for future developments. Comput. Ind. **57**(6), 490–503 (2006)
3. Ruffo, M., Tuck, C., Hague, R.: Make or buy analysis for rapid manufacturing. Rapid Prototyp. J. **13**(1), 23–29 (2007)
4. Atzeni, E., Salmi, A.: Economics of additive manufacturing for end-usable metal parts. Int. J. Adv. Manuf. Technol. **62**(9–12), 1147–1155 (2012)
5. Thiesse, F., Wirth, M., Kemper, H.-G., Moisa, M., Morar, D., Lasi, H., Piller, F., Buxmann, P., Mortara, L., Ford, S., et al.: Economic implications of additive manufacturing and the contribution of mis. Bus. Inf. Syst. Eng. **57**(2), 139 (2015)
6. Holmström, J., Partanen, J., Tuomi, J., Walter, M.: Rapid manufacturing in the spare parts supply chain: alternative approaches to capacity deployment. J. Manuf. Technol. Manag. **21**(6), 687–697 (2010)
7. Khajavi, S.H., Partanen, J., Holmström, J.: Additive manufacturing in the spare parts supply chain. Comput. Ind. **65**(1), 50–63 (2014)
8. Liu, P., Huang, S.H., Mokasdar, A., Zhou, H., Hou, L.: The impact of additive manufacturing in the aircraft spare parts supply chain: supply chain operation reference (scor) model based analysis. Prod. Plan. Control **25**(13–14), 1169–1181 (2014)
9. Wullms, B., Arts, J.J., Topan, E.E., van der Schoot, H., Schindler-Chaloub, J.: Additive manufacturing in the spare parts supply chain. Eindhoven Univ. Tech. **11**, 85 (2014)
10. Moinzadeh, K., Schmidt, C.P.: An (s- 1, s) inventory system with emergency orders. Oper. Res. **39**(2), 308–321 (1991)
11. Vlachos, D., Tagaras, G.: An inventory system with two supply modes and capacity constraints. Int. J. Prod. Econ. **72**(1), 41–58 (2001)
12. Teunter, R., Vlachos, D.: An inventory system with periodic regular review and flexible emergency review. IIE Trans. **33**(8), 625–635 (2001)
13. Sheopuri, A., Janakiraman, G., Seshadri, S.: New policies for the stochastic inventory control problem with two supply sources. Oper. Res. **58**(3), 734–745 (2010)
14. Gong, X., Chao, X., Zheng, S.: Dynamic pricing and inventory management with dual suppliers of different lead times and disruption risks. Prod. Oper. Manag. **23**(12), 2058–2074 (2014)
15. Zhu, S.X.: Analysis of dual sourcing strategies under supply disruptions. Int. J. Prod. Econ. **170**, 191–203 (2015)
16. Yang, J., Qi, X., Xia, Y.: A production-inventory system with markovian capacity and outsourcing option. Oper. Res. **53**(2), 328–349 (2005)
17. Jakšič, M., Fransoo, J.: Dual sourcing in the age of near-shoring: trading off stochastic capacity limitations and long lead times (under revision)

Drivers and Resistors for Supply Chain Collaboration

Verena Jung, Marianne Peeters and Tjark Vredeveld

Abstract Due to a constantly growing competition among organizations and higher customer expectations, in the last decades companies started to realize the need for supply chain collaboration (SCC). Although the idea of SCC may sound easy in theory, SCCs in practice often fail. This can be explained by the fact that for a specific SCC a huge amount of drivers and resistors has to be taken into account by all parties involved. However, these drivers and resistors are often unknown or misunderstood by the parties, which leads to the fact that SCCs likely fail. To avoid this, we present a framework which provides a complete overview and the correct understanding of all possible drivers and resistors identified through an extensive literature review. The completeness of the framework in practice is investigated through interviews with companies. In theory, usually dyadic relationships are observed. The companies we interviewed participated in triangular relationships or in SCCs where even more than three parties were involved. Preliminary results indicate that even for these more complex relationships in practice the framework is complete.

1 Introduction

In the last decades companies realized the need for looking outside their organizational boundaries for new opportunities [4, 10, 12]. According to [9, 13], a new vital base of competitive advantage that has not yet been fully exploited is supply chain collaboration (SCC). Nowadays, SCC is a widely discussed topic and

V. Jung (✉) · M. Peeters · T. Vredeveld
Maastricht University, Maastricht, The Netherlands
e-mail: v.jung@maastrichtuniversity.nl

M. Peeters
e-mail: m.peeters@maastrichtuniversity.nl

T. Vredeveld
e-mail: t.vredeveld@maastrichtuniversity.nl

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_82

it means that "two or more independent companies work jointly to plan and execute [...] operations with greater success than when acting in isolation" [12].

Although the idea of SCC may sound easy in theory, SCCs in practice often fail. According to [6], SCC promises theoretically huge benefits but it appears that reality falls short, which indicates a gap between theory and practice. This can be explained by the fact that for a specific SCC a huge amount of drivers and resistors has to be taken into account by all parties involved. However, these drivers and resistors are often unknown or misunderstood by the parties, which leads to the fact that SCCs likely fail. To avoid this, it is necessary that the parties have a complete overview of all possible drivers and resistors, so that the parties can identify their relevant drivers and resistors for the specific SCC. To achieve this, it is important that the parties have the correct understanding of the different drivers and resistors.

The goal of our paper is to create a framework, which should provide this complete overview and to ensure that the parties are provided with the correct understanding of all drivers and resistors. Until now, such a framework is missing in literature. The remainder of the paper is organized as follows. In Sect. 2 a critical discussion of an extensive literature review is given, and based on this our framework is created. Next, the completeness of the framework in practice. Preliminary results are presented in Sect. 3. Furthermore, we intend to investigate whether the framework can provide parties with a higher awareness of all possible drivers and resistors in order to increase the probability of successful SCCs and to close the gap between theory and practice. Ideas are presented in an outlook in Sect. 3 as well.

2 Critical Discussion and Framework

Until now, a great amount of researchers tried to identify important drivers and resistors for SCC but there exists some ambiguity in literature. Through an extensive literature review, two different kinds of ambiguity for the drivers could be identified. The first kind of ambiguity is that the same terms are often used for different categories and that for each category no unique term exists. Examples are the terms "drivers" and "driving forces". In [1] the term "drivers" is used to define two different categories. First, they use the term to define factors which enable someone to collaborate like "trust" or "commitment". Second, they use the term for expected benefits of a successful SCC like "enhancement in customer service" or "increase in market share". Next to the term "drivers" they also use the term "driving forces" for the expected benefits. However, the term "driving forces" is used by [8] to define factors which force a party to collaborate like "more demanding customers" or "economic globalization". The second kind of ambiguity is that factors are assigned to more than one category. An example is the factor "trust", which is assigned in [1] to the factors which enable someone to collaborate and in [3] to the outcomes of SCC. For the resistors a unique term and definition is also still missing. However, in contrast to the drivers most of the time only one category is named for the resistors. Even for this single category there exists different terms like "barriers" (e.g. [2]) or

"impediments" (e.g. [5]). Moreover, for the drivers and resistors it can be observed that in one paper a factor of a category is named but is missing in another paper. An example is the factor "interdependence" which is named as a factor which enables someone to collaborate in [1] but it is not mentioned in [2] for the same category.

Because of the ambiguity, it is hard for the parties to understand and identify the relevant drivers and resistors for their specific SCC. Therefore, a clear structure and a complete overview of the drivers and resistors is needed, which is provided by our following framework.

Our framework consists of two umbrella terms for the categories. The first umbrella term is called "drivers" and the second one is called "resistors". We choose the term "drivers" because it has been used ambiguous for every category of the drivers in the literature. For all the retarding factors we use the umbrella term "resistors" because it represents the meaning very well. The drivers are divided up into three different categories. The first category, "benefits", represents the expected benefits of a successful SCC like "cost reduction" (e.g. [2]). The second category, "forces", and it represents external factors which force a company to collaborate like "greater competitive intensity" (e.g. [8]). The last category, "enablers", represents the factors which enable someone to collaborate and, in addition, have an effect on the success of SCC like "trust" (e.g. [1]). The distinction between the three categories has been made because they all represent different drivers and, additionally, have different influences on SCC. Benefits and forces are both motivation factors for parties to collaborate. Nevertheless, there is a big difference. Benefits represent the intrinsic motivation, which means that the party decides to collaborate out of its own motivation [10]. Therefore, the benefits usually have a positive influence on SCC. However, forces represent the extrinsic motivation. Here a party is forced to collaborate. This has a great effect on management's ability to implement SCC which can have a negative influence on SCC, because a change in management practices towards more SCC is dictated but not necessarily wished by the company itself [8]. It is important to understand that not only resistors can have a negative influence on collaboration, but also a category of the drivers, the forces. In addition to the two kinds of motivation, enablers are needed, because a strong desire to build a SCC is not enough. The enablers increase the probability of success and, therefore, they have a positive influence on SCC [10]. The resistors are divided up into two categories. One category is called "barriers" and this are impediments, which could limit SCC before collaborating. An example is "lack of commitment" (e.g. [2]). The other category is called "risks". Risks are events which might occur in the future but they are unknown yet like "decreased competitiveness" (e.g. [11]). This distinction has not often been made in literature so far. However, it is necessary to distinguish between these two categories because a barrier is something which is occurring now and a risk is future-based. This was also mentioned in [7] in the context of strategic planning. Both categories of the resistors have a negative influence on SCC. However, there is a difference, because the barriers are already known for sure when the party decides whether to collaborate or not. This leads to the fact that the party can already take actions against the barriers. In contrast to this, the risks are not known for sure at this point in time, only an uncertainty that something might occur can be present.

Because of this, the decision to collaborate or not is dependent on the decision maker and his risk preference (risk averse, risk neutral or risk loving).

The current framework represents what we have found in the literature. However, we want to go one step further because we saw that there is a connection between the driver "enablers" and the resistor "barriers". It is striking, that for every enabler a resistor, in particular a corresponding barrier, can be found, which is stressed by [15]. An example is the enabler "trust" and the barrier "lack of trust" (e.g. [14]). Therefore, we combine the factors of these two categories. The factors can either be present in a collaboration, "presence of ...", or not, "lack of ...".

In Fig. 1 our framework with definitions and explanations of the influences of the categories on SCC together with some examples of different factors for each category is presented. To get a better overview we group the factors in an intuitive way. An example is that the factor "cost reduction" is a general term for e.g. transportation or inventory cost reduction.

To evaluate a potential SCC, it is important for parties to have this complete overview given. However, it is not necessarily the case that all factors, except from the category "enablers/barriers", are relevant in every SCC; it depends on the party itself, its industry and the type of SCC. An example is provided by [2] who investigated a SCC in the construction industry from the manufacturer's perspective. In their paper they identified "reducing bureaucracy and paperwork" as an important benefit. This is a direct benefit for the manufacturer and, therefore, a relevant benefit for him but probably not for a logistic service provider (LSP). Moreover, it is possible that a whole category is not relevant for a SCC. When a party is forced to collaborate the intrinsic motivation for starting this SCC, so all factors in this category, can be absent.

In summary, a complete overview and the right understanding of all possible factors is needed, which is provided by our framework. Without the framework it is often not possible to identify all relevant factors. However, missing an important one can increase the probability of failure.

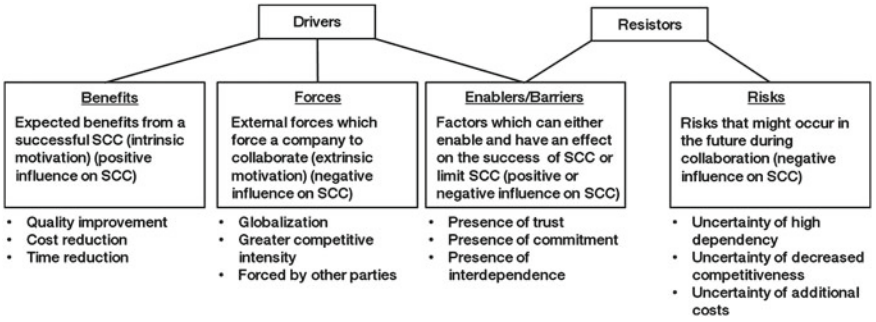


Fig. 1 Our framework

3 Preliminary Results and Outlook

Our main goal in the research is to investigate whether our literature based framework is also complete in practice. Hereto, we conducted individual semi-structured in-depth interviews with manufacturers, LSPs and retailers from the Dutch fast moving consumer goods industry. The purpose of the interviews is to investigate whether relevant general factors are missing, through the identification of important factors for parties to start a SCC. In theory, usually dyadic relationships are observed. However, the companies we interviewed participated mostly in triangular relationships or in SCCs with even more than three involved parties. A triangular relationship is more complex than a dyadic, because instead of two parties three parties try to start a SCC. Therefore, also three parties have to identify their relevant factors for starting a SCC taking into account the other two parties, which increases the complexity. Hence, if no additional general factors can be identified, we can conclude that the framework is complete for triangular but also for less complex dyadic relationships. Preliminary results indicate that no additional general factor is needed.

Our secondary goal is to investigate whether the framework can increase the probability of successful SCCs and, thus, close the gap between theory and practice by providing a higher awareness of all possible factors. This includes an increase in the likelihood that SCCs with a high failure probability will not be started. Results to that research question cannot be presented yet. However, our intention is to interview parties who participated in a failed SCC to investigate whether the failure would have been prevented if the party had given our framework. To increase the reliability of our research, we intend to cluster the factors for every specific SCC.

In summary, we showed that there exists some degree of ambiguity in the literature that makes the identification and the correct understanding of the different drivers and resistors for staring SCC nearly impossible. This leads to the fact that SCCs often fail. To avoid this, we presented a framework which provides a complete overview and the correct understanding of all possible drivers and resistors. In our research we have been investigating whether the framework which covers the existing literature is complete in practice. Preliminary results indicate this. The investigation whether the framework can increase the probability of a successful SCC is what we will do next. Furthermore, future research will investigate what influence Stackelberg games have on the relevant factors in our framework.

References

1. Ahmad, S., Ullah, A.: Driving forces of collaboration in supply chain: a review. Interdisc. J. Contemp. Res. Bus. **5**, 39–69 (2013)
2. Akintoye, A., McIntosh, G., Fitzgerald, E.: A survey of supply chain collaboration and management in the UK construction industry. Eur. J. Purchasing Supply Manag. (2000). doi:10.1016/S0969-7012(00)00012-5
3. Beach, R., Webster, M., Campbell, K.M.: An evaluation of partnership development in the construction industry. Int. J. Proj. Manag. (2005). doi:10.1016/j.ijproman.2005.04.001

4. Cao, M., Zhang, Q.: Supply chain collaboration: Impact on collaborative advantage and firm performance. J. Oper. Manag. (2011). doi:10.1016/j.jom.2010.12.008
5. Cruijssen, F., Dullaert, W., Fleuren, H.: Horizontal cooperation in transport and logistics: a literature review. Transp. J. **46**, 22–39 (2007)
6. Daugherty, P.J., Richey, R.G., Roath, A.S., et al.: Is collaboration paying off for firms? Bus. Horiz. (2006). doi:10.1016/j.bushor.2005.06.002
7. Evans, J.: Managing assumptions, risks and impediments in strategic planning. executive street: the business leader's resource (2012). http://blog.vistage.com/business-strategy-and-management/managing-assumptions-risks-and-impediments-in-strategic-planning/. Accessed 13 Oct 2012
8. Fawcett, S.E., Magnan, G.M., McCarter, M.: A three-stage implementation model for supply chain collaboration. J. Bus. Logist. (2008). doi:10.1002/j.2158-1592.2008.tb00070.x
9. Horvath, L.: Collaboration: the key to value creation in supply chain management. Int. J. Supply Chain Manag. (2001). doi:10.1108/EUM0000000006039
10. Lambert, D.M., Emmelhainz, M.A., Gardner, J.T.: Developing and implementing supply chain partnerships. Int. J. Logist. Manag. (1996). doi:10.1108/09574099610805485
11. Maloni, M.J., Benton, W.C.: Supply chain partnerships: Opportunities for operations research. Eur. J. Oper. Res. (1997). doi:10.1016/s0377-2217(97)00118-5
12. Simatupang, T.M., Sridharan, R.: The collaborative supply chain. Int. J. Logist. Manag. (2002). doi:10.1108/09574090210806333
13. Sukati, I., Hamid, A.B.A., Baharun, R., et al.: Competitive advantage through supply chain responsiveness and supply chain integration. Int. J. Bus. Commer. **1**, 1–11 (2012)
14. Visser, L.J.: Thresholds in logistics collaboration decisions: a study in the chemical industry. Dissertation, Tilburg University (2010)
15. Walker, H., Di Sisto, L., McBain, D.: Drivers and barriers to environmental supply chain management practices: lessons from the public and private sectors. J. Purchasing Supply Manag. (2008). doi:10.1016/j.pursup.2008.01.007

Balancing Effort and Plan Quality: Tactical Supply Chain Planning in the Chemical Industry

Annika Vernbro, Iris Heckmann and Stefan Nickel

Abstract Tactical supply chain planning in the chemical industry is a frequently recurring task, which typically involves not only automated planning procedures but also manual planning efforts. Competent application of optimization techniques in this context requires specialized skills and method-related knowledge. However, planners might have a background in overseeing production or comparable functions and personnel with expertise in optimization is scarce. Common simple heuristic planning approaches provided by ERP-systems do not even consider capacity constraints. This leads to considerable amounts of manual planning efforts. Still, for complex supply chains the resulting plans likely lack in quality. Available advanced heuristics can produce better plans but at the same time require expertise and maintenance efforts comparable to optimization based approaches. We introduce a concept of quality of supply network plans and investigate the potential for achieving balance between effort and plan quality by incorporating certain dimensions of quality in application friendly planning heuristics.

1 Introduction

Models for tactical supply chain planning from literature make implicit assumptions on plan quality requirements by incorporating certain constraints, choosing an objective function and choosing a level of data- and model-granularity. For chemical and related industries this concerns basic mixed integer programs (as e.g. in [12]) as well as advanced models (as e.g. in [1, 3, 4, 10]). To the best of our knowledge there is not

A. Vernbro (✉) · I. Heckmann
FZI Research Center for Information Technology, Haid-und-Neu Str. 10-14,
76131 Karlsruhe, Germany
e-mail: vernbro@fzi.de

I. Heckmann
e-mail: heckmann@fzi.de

S. Nickel
Karlsruhe Institute of Technology, Englerstr. 11, 76131 Karlsruhe, Germany
e-mail: stefan.nickel@kit.edu

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_83

yet an explicit definition of what constitutes quality of tactical plans. [2] stresses the importance of integration and comprehensiveness in planning to enable creation of value in the supply chain, [9] emphasizes linking planning levels in a planning hierarchy. According to our insights from planning practice in the chemical industry, tactical plans often lack utility for subsequent planning steps and dependent functional areas. Reevaluation of common implicit assumptions on plan quality is necessary. At the same time advanced planning methods may be rejected in practice due to usability issues.

We discuss the potential for certain aspects of plan quality to be considered by a still simple heuristic planning approach and evaluate a heuristic planning algorithm for this purpose based on a real instance from chemical industry.

2 Conceptual Approach to Tactical Plan Quality

The planning process is designed to enable operations which meet demand in the intended way. This concerns cost minimization/profit maximization objectives as well as preference of certain customers or markets according to strategic goals. Within the hierarchical planning process (see [5, 11]) tactical planning performs a preparatory and coordinating role. This includes properly preparing and coordinating subsequent planning steps (like scheduling or procurement planning) under consideration of industry specific features like sequence dependent changeovers. We derive:

Definition 1 *The quality of a tactical supply chain plan* is determined by the degree to which it (via subsequent planning steps) enables operations to match demand in the way intended by business strategy. Core aspects are:

1. *Fulfillment of objectives set by business strategy* (like cost minimization/profit maximization/... or prioritization of certain demands) evaluated based on detailed supply chain information.
2. *Appropriate preparation and coordination of subsequent planning steps*.
3. *Consideration of operations-specific requirements, which concern the tactical horizon* (like aspects of similarity between subsequent plans).

The first core aspect is to be treated with care. The degree to which strategic goals like profit maximization are met by an aggregate tactical plan reveals itself only under consideration of detailed supply chain data and information—particularly in case of substantial sequence dependent changeovers.

The degree to which plans generated in tactical planning meet the postulated quality requirements depends on the choice of planning method. At the same time this choice affects planning efforts and complexity. With regard to planning practice we suggest to consider these aspects jointly when choosing or designing a planning method.

3 A Simple Heuristic in the Light of Plan Quality

Plan quality as conceptualized in Sect. 2 can be influenced by means of the complexity and sophistication of planning models and methods. At the same time these levers have an impact on planning efforts.

Based on the uncapacitated algorithm we encountered in planning practice we briefly describe a basic heuristic planning approach for Supply Network Planning (SNP). It incorporates freely available information on the uncapacitated heuristic algorithm as implemented in the SAP APO Supply Network Planning-module (see [7]).

Principles of the BasicUncapacitated-Heuristic

1. Move upstream through the tiers of the supply chain, for each tier move through production facilities following predefined order
2. In case of multiple sourcing options, initially assign dependent demands to facilities of following tiers according to predefined quota
3. Starting from primary demands, following the order from (1.) plan production quantities and derive dependent demands via given bills of material without consideration of capacity constraints.

Incorporating capacity constraints in the planning algorithm can save large amounts of manual replanning efforts to arrive at a feasible plan while improving plan quality.

Incorporating capacity constraints and quality requirements: BasicPQ-Heuristic
In the following we discuss whether central aspects of plan quality can additionally be incorporated in a still simple planning heuristic like the *BasicUncapacitated*-heuristic. Based on the core aspects of tactical plan quality as introduced in Sect. 2 we consider a selection of concrete quality requirements:

(a) Profit maximization
(b) Prioritization of demands
(c) Issues with preparation and coordination of subsequent planning steps related to e.g. sequence dependent changeovers
(d) Similarity over time horizons.

In Fig. 1 we introduce a planning heuristic for SNP which extents the *BasicUncapacitated* heuristic described above by consideration of production capacities and prioritization of product-demands e.g. based on profit margins. Capacity constraints are enforced based on the priority ranking of the primary demands which dependent demands at a certain facility originate from. Due to shelf life-issues in the chemical industry, cuts regarding ingredients for a certain end product are propagated for all other ingredients.

For initial illustration, here, we compare the results of BasicPQ under two different demand allocation quotas and a profit-optimal assignment for a small, simplified, two-period instance without backlog (see Fig. 2). We observe, that already

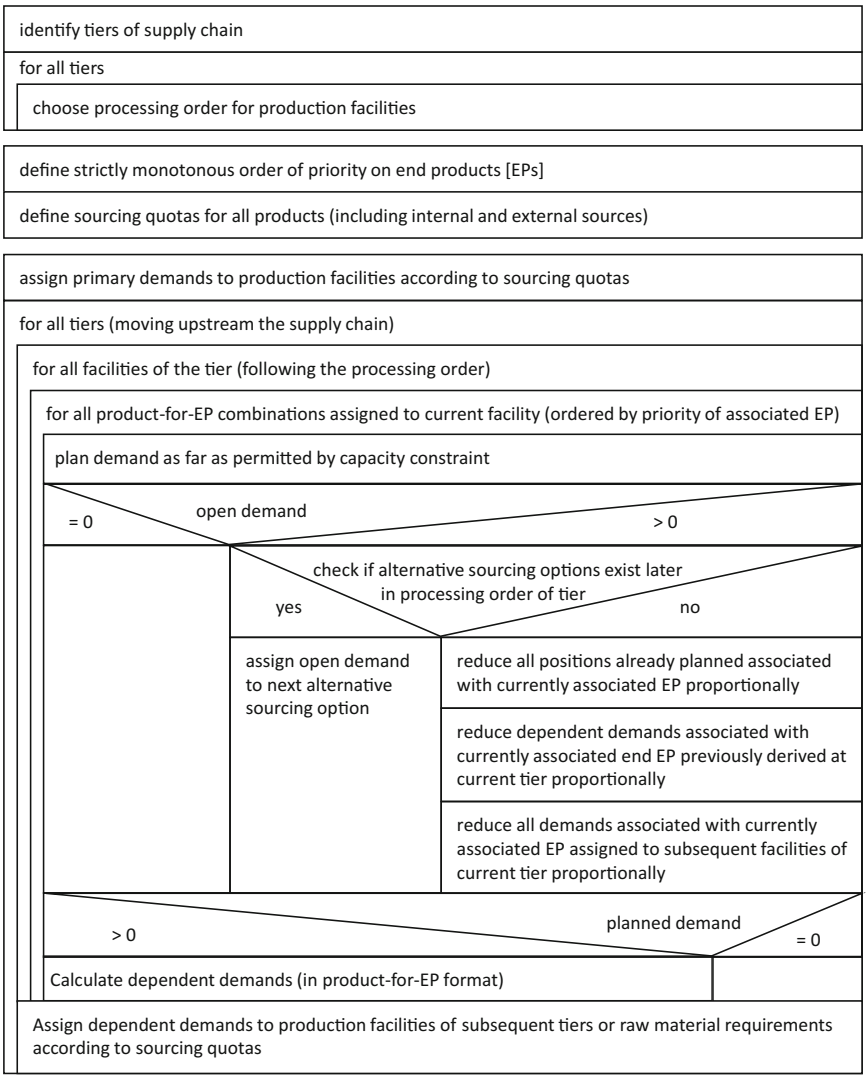
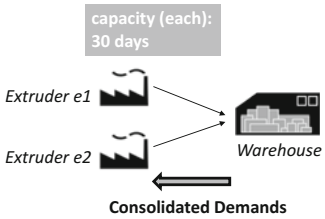


Fig. 1 Overview BasicPQ-heuristic

for this very small instance the results of the heuristic approach show a variation in deviation from an optimal plan depending on the demand figures and demand allocation quotas. For this small, exemplary instance different quotas even allow the BasicPQ-Heuristic to deliver profit-optimal plans for the different demand instances encountered in period 1 and 2 (see Fig. 2).



Location Products with Capacity Consumption in days per Base Unit	Extruder e1	Extruder e2
Product A	1	2
Product B	2	1
Product C	1	1

Profit Per Base Unit of Product (in Monetary Units)	
Product A	10
Product B	9
Product C	1

Demand Allocation Quota for BasicPQ-Heuristic	Strategy S1		Strategy S2	
	e1	e2	e1	e2
Prod A	1	0	1	0
Prod B	0	1	1	0
Prod C	0.5	0.5	1	0

Consolidated Demands in Base Units	Period 1	Period 2
Prod A	26	26
Prod B	35	26
Prod C	10	10

Production Plans

		Period 1 A	B	C	Profit	Period 2 A	B	C	Profit
S1	e1	26	0	4	534	26	0	4	502
	e2	0	30	0		0	26	4	
S2	e1	26	2	0	548	26	2	0	484
	e2	0	30	0		0	22	8	
(Profit-) optimal Plan	e1	26	2	0	548	26	0	4	502
	e2	0	30	0		0	26	4	

Fig. 2 Comparison of BasicPQ-heuristic with prioritization of demands based on profit per unit under different demand allocation quotas and profit-optimal assignment for a rudimentary instance

Aspects of Quality-Performance

The *BasicPQ*-heuristic was evaluated based on an appropriately modified two-tier instance from chemical industry (ten processing- and three filling-facilities, 128 products with 686 facility-specific bills of material) and benchmarked against a basic MIP-formulation for SNP at the same level of detail and information. Exploratory analyses for ten demand instances suggest a large influence of the chosen demand allocation quota on achievable profit (see quality requirement (*a*), (*b*)). With the best found quota-assignment-strategy *BasicPQ* reached on average 78% of the MIP-benchmark-value. Facility assignment per material for different demand instances was stable which is favorable in terms of similarity requirements (see quality requirement (*d*)).

We expect overall and detailed network structure (e.g. for serial, divergent and convergent structures see [8]) to impact the performance potential of the *BasicPQ*-heuristic. As demonstrated in [13] the utility of tactical plans based on aggregate simplified supply chain information (as *BasicPQ* is based on) is strongly dependent on the degree of sequence dependency and extent of change over times (see quality requirement (*c*)). Additionally, suitable campaign lengths with regard to inventory holding and changeover costs should impact the applicability of the *BasicPQ*-heuristic. As a next step we suggest sensible utilization of idle capacity based on inventory levels. Extent and effect of uncertainty regarding demand and supply chain operations (see [6]) should be investigated as well.

4 Conclusions

Motivated by practicability issues in the chemical industry regarding advanced planning methods for tactical supply chain planning in practice, we suggested balancing planning efforts and plan quality and proposed a novel structural approach to plan quality. We enriched a customary simple uncapacitated heuristic planning approach by considering certain aspects of plan quality. The *BasicPQ*-heuristic was illustrated by a numeric example, and evaluated based on a real instance from chemical industry. Theoretical deliberations already make it obvious, that a simple approach of this type can only be sensibly applied for supply chains with certain characteristics, particularly regarding sequence dependent changeovers and campaign lengths.

References

1. Dansereau, L.P., El-Halwagi, M., Mansoornejad, B., Stuart, M.: Framework for margins-based planning: forest biorefinery case study. Comput. Chem. Eng. **63**, 34–50 (2014)
2. Goetschalckx, M.: Supply Chain Engineering. Springer, New York, Dordrecht, Heidelberg, London (2011)
3. Grunow, M., Guenther, H.-O., Lehmann, M.: Campaign planning for multi-stage batch processes in the chemical industry. OR Spectr. **24**, 281–314 (2002)
4. Grunow, M., Guenther, H.-O., Yang, G.: Plant co-ordination in pharmaceutics supply networks. OR Spectr. **25**, 109–141 (2003)
5. Hax, A.C., Meal, H.C. (1973). Hierarchical integration of production planning and scheduling. Massachusetts Institute of Technology (MIT). Sloan School of Management. http://ideas.repec.org/p/mit/sloanp/1868.html. Accessed 02 Aug 2016
6. Heckmann, I., Comes, T., Nickel, S.: A critical review on supply chain risk—definition, measure and modeling. Omega **52**, 119–132 (2015)
7. Heuristikbasierte Planung. http://help.sap.com/
8. Meyr, H., Stadtler, H.: Types of Supply Chains. In: Kilger, C. (ed.) Supply Chain Management and Advanced Planning, 4th edn. Springer, Berlin (2008)
9. Schneeweiss, C.: Distributed Decision Making, 2nd edn. Springer, Berlin (2003)
10. Sel, C., Bilgen, B., Bloemhof-Ruwaard, J.M., van der Vorst, J.G.A.J.: Multi-bucket optimization for integrated planning and scheduling in the perishable dairy supply chain. Comput. Chem. Eng. **77**, 59–73 (2015)
11. Stadtler, H., Fleischmann, B.: Hierarchical planning and the supply chain planning matrix. In: Stadtler, H., Fleischmann, B., Grunow, M., Meyr, H., Suerie, C. (eds.) Advanced Planning in Supply Chains, 1st edn. Springer, Berlin (2012)
12. Stadtler, H.: Master planning—supply network planning. In: Stadtler, H., Fleischmann, B., Grunow, M., Meyr, H., Suerie, C. (eds.) Advanced Planning in Supply Chains, 1st edn. Springer, Berlin (2012)
13. Vernbro, A., Heckmann, I., Nickel, S.: PrO-lean planning: detail and aggregation in tactical supply chain planning for the chemical industry. In: Talk at the 27th European Conference on Operational Research (2015)

Part XXI
Traffic and Passenger Transportation

The Modulo Network Simplex with Integrated Passenger Routing

Ralf Borndörfer, Heide Hoppmann, Marika Karbstein and Fabian Löbel

Abstract Periodic timetabling is an important strategic planning problem in public transport. The task is to determine periodic arrival and departure times of the lines in a given network, minimizing the travel time of the passengers. We extend the modulo network simplex method (Nachtigall and Opitz, Solving periodic timetable optimisation problems by modulo simplex calculations 2008 [6]), a well-established heuristic for the periodic timetabling problem, by integrating a passenger (re)routing step into the pivot operations. Computations on real-world networks show that we can indeed find timetables with much shorter total travel time, when we take the passengers' travel paths into consideration.

1 Introduction

Classical optimization approaches to periodic timetabling are based on formulations in terms of the period event scheduling problem (PESP) [7], see, e.g., Liebchen [4] and the references therein. A powerful heuristic for the PESP is the *modulo network simplex method*, which has been proposed by Nachtigall and Opitz [6]. It iteratively improves a given feasible solution by pivot operations. This algorithm has been improved by Goerigk and Schöbel [3] who introduced pivot selection rules and cuts to escape local optima.

This research was carried out in the framework of MATHEON supported by Einstein Foundation Berlin.

R. Borndörfer · H. Hoppmann (✉) · M. Karbstein · F. Löbel
Zuse Institute, Takustr. 7, 14195 Berlin, Germany
e-mail: hoppmann@zib.de

R. Borndörfer
e-mail: borndoerfer@zib.de

M. Karbstein
e-mail: karbstein@zib.de

F. Löbel
e-mail: fabian.loebel@zib.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_84

Standard PESP models work with fixed travel paths. The passengers, however, choose their routes depending on the timetable. Approaches to integrate passenger routing in periodic timetabling have been presented recently, see, e.g., [1, 2]. In this paper, we propose to apply the modulo network simplex method to a periodic timetabling model with variable passenger routing, i.e., to the integrated periodic timetabling and passenger routing problem. We show that a pivot selection that considers updated passenger routes allows to find better timetables in terms of total travel time.

2 Periodic Timetabling with Fixed Passenger Routes

Consider a directed graph $N = (V, A)$, the *event-activity network*. The nodes V are called *events* and represent arrivals and departures of lines at their stations. The arcs $A \subseteq V \times V$ model *activities* of lines (driving between stations, waiting at stations) and possible transfers between lines at stations. Further, we are given lower and upper time bounds $\ell_a, u_a \in \mathbb{Q}_{\geq 0}$, respectively, for the duration of activity $a \in A$. Activity weights $w \in \mathbb{R}^A_{\geq 0}$ represent the number of the passengers traveling on arc $a \in A$.

A *periodic timetable* $\pi : V \to \mathbb{Q}$ determines for each line periodic arrival and departure times at its stations. We call a timetable *feasible* if π satisfies the *periodic interval constraints* $\ell_a \leq [\pi_j - \pi_i]_T \leq u_a$ for each activity $a = (ij) \in A$; here, we define $[y]_T := y \mod T$ for $y \in \mathbb{R}$. We may assume without loss of generality that $0 \leq \ell_a \leq u_a$, $u_a - \ell_a < T$, and $\ell_a < T$ holds for all $a \in A$, see [4]. By Serafini and Ukovich [7], π satisfies the periodic interval constraints if and only if there exist *modulo parameters* $z \in \mathbb{Z}^A$ such that $\ell_a \leq \pi_j - \pi_i + T z_a \leq u_a \, \forall \, a = (ij) \in A$. For a feasible timetable π with modulo parameters z, the resulting duration of activity $a = (ij) \in A$ is given by $x_a := \pi_j - \pi_i + T z_a$, and is called *periodic tension*. The *periodic slack* is defined by $y_a := x_a - \ell_a$; it measures how much the lower bound is exceeded. The goal is to find a feasible timetable such that the resulting weighted total travel time of all passengers is minimized.

Periodic tensions and slacks can be characterized by means of cycles in N, see [4, 5]. Let $\mathcal{T} \subseteq A$ be a spanning tree of N. For a co-tree arc $\bar{a} \in A \setminus \mathcal{T}$, denote by $C_{\bar{a}}$ the *fundamental cycle of* $\bar{a}$, i.e., the unique oriented cycle $C_{\bar{a}}$ induced by adding $\bar{a}$ to the tree. Arcs in $C_{\bar{a}}$ with the same orientation as $\bar{a}$ are called forward arcs $C^+_{\bar{a}}$, arcs with opposite orientation are called backward arcs $C^-_{\bar{a}}$. The *fundamental cycle matrix* $\Gamma \in \{-1, 0, 1\}^{A \setminus \mathcal{T} \times A}$ of $\mathcal{T}$ is defined by $\Gamma_{\bar{a}a} = 1$ if $a \in C^+_{\bar{a}}$, $\Gamma_{\bar{a}a} = -1$ if $a \in C^-_{\bar{a}}$, and $\Gamma_{\bar{a}a} = 0$ if $a \notin C_{\bar{a}}$ for all $\bar{a} \in A \setminus \mathcal{T}$ and $a \in A$.

We introduce slack variables $y \in \mathbb{Q}^A$ for the arcs and modulo parameter variables $z \in \mathbb{Z}^{A \setminus \mathcal{T}}$ for the co-tree arcs of $\mathcal{T}$. As suggested by Nachtigall [5], the periodic timetabling problem can be formulated as the following integer program:

$$(\mathrm{PTT}_w) \quad \min \quad \sum_{a\in A} w_a\,(y_a + \ell_a)$$

$$\text{s.t.} \quad \Gamma\, y - T\, z = -\Gamma\, \ell \tag{1}$$

$$0 \le y_a \le u_a - \ell_a \qquad \forall\, a \in A \tag{2}$$

$$y_a \in \mathbb{Q} \qquad \forall\, a \in A \tag{3}$$

$$z_a \in \mathbb{Z} \qquad \forall\, a \in A \setminus \mathcal{T}. \tag{4}$$

The model (PTT_w) minimizes the total passenger travel time for a fixed passenger routing given by the arc weights $w \in \mathbb{R}^A_{\ge 0}$. A timetable given by tensions is feasible if and only if the tensions sum up to a multiple of the period time along every fundamental cycle. This is expressed by Eq. (1) in terms of slack variables.

3 The Modulo Network Simplex Method

In this section, we recall the modulo network simplex method as proposed by Nachtigall and Opitz [6].

A point $(y, z) \in \mathbb{R}^A \times \mathbb{Z}^{A\setminus\mathcal{T}}$ is called a *spanning tree solution* for (PTT_w), if there exists a spanning tree structure $\mathcal{S} = \mathcal{S}_\ell \cup \mathcal{S}_u$, where $\mathcal{S}$ is a spanning tree of N, the periodic slack y_a is zero for all $a \in \mathcal{S}_\ell$, and at its upper bound $u_a - \ell_a$ for all $a \in \mathcal{S}_u$. The values for all non-tree arcs and the modulo parameters are uniquely determined by Eq. (1). The spanning tree solution is called feasible if $0 \le y_a \le u_a - \ell_a$ for all $a \in A$, i.e., (y, z) is a feasible solution of (PTT_w).

Theorem 1 (Nachtigall [5]) *Define the* periodic slack polyhedron *by*

$$\mathcal{Y} := \operatorname{conv}\left\{(y, z) \in \mathbb{R}^A \times \mathbb{Z}^{A\setminus\mathcal{T}} : 0 \le y \le u - \ell, \Gamma\, y - T\, z = -\Gamma\, \ell\right\}.$$

Then, $(y, z) \in \mathcal{Y}$ is an extremal point of $\mathcal{Y}$ if and only if it is a spanning tree solution.

The idea of the modulo network simplex is as follows: starting with a feasible spanning tree solution (y, z) for a spanning tree structure $\mathcal{S} = \mathcal{S}_\ell \cup \mathcal{S}_u$, the current solution is iteratively improved by exchanging a co-tree arc $\bar{a} \in A \setminus \mathcal{S}$ with a tree arc $\hat{a} \in \mathcal{S}$ in its fundamental cycle. This is done by shifting the slack from the co-tree arc $\bar{a}$ to the other arcs in the fundamental cut of $\hat{a}$. For every tree arc $\hat{a} \in \mathcal{S}$, the *fundamental cut induced by* $\hat{a}$ is defined by the unique minimal oriented cut $\mathcal{X}_{\hat{a}} \subseteq A$ of N such that $\mathcal{X}_{\hat{a}} \cap \mathcal{S} = \hat{a}$. As commonly known, $\bar{a}$ is contained in the fundamental cut induced by $\hat{a}$ if and only if $\hat{a}$ is contained in the fundamental cycle induced by $\bar{a}$.

Let $\tilde{\Gamma}$ be the fundamental cycle matrix of $\mathcal{S}$ and let $\delta \in \{y_{\bar{a}}, y_{\bar{a}} - u_{\bar{a}} + \ell_{\bar{a}}\}$. Then

$$y'_a = \begin{cases} [y_a + \tilde{\Gamma}_{\bar{a}\hat{a}}\,\delta]_T & \text{if } a \in \mathcal{X}^+_{\hat{a}}, \\ [y_a - \tilde{\Gamma}_{\bar{a}\hat{a}}\,\delta]_T & \text{if } a \in \mathcal{X}^-_{\hat{a}}, \\ y_a & \text{else,} \end{cases} \qquad \forall\, a \in A,$$

induces a feasible spanning tree solution if $y'_a \le u_a - \ell_a$ for all $a \in A$. That is, if $y'_a \le u_a - \ell_a$ for all $a \in A$, then there exists $z' \in \mathbb{Z}^{A \setminus \mathcal{T}}$ such that (y', z') is a feasible spanning tree solution of (PTT_w) with respect to $\mathcal{S}' = \mathcal{S} \cup \{\bar{a}\} \setminus \{a\}$. If $\delta = y_{\bar{a}}$, then we are pivoting the co-tree arc $\bar{a}$ into $\mathcal{S}'_\ell$, i.e., $y'_{\bar{a}} = 0$. On the other hand, if $\delta = y_{\bar{a}} - u_{\bar{a}} + \ell_{\bar{a}}$, then we are pivoting the co-tree arc $\bar{a}$ into $\mathcal{S}'_u$, i.e., $y'_{\bar{a}} = u_{\bar{a}} - \ell_{\bar{a}}$.

We call y' a *feasible pivot operation* if y' is a feasible solution. If the difference in the objective value is negative, i.e., $\sum_{a \in A} w_a (y'_a + \ell_a) < \sum_{a \in A} w_a (y_{a+\ell_a})$, then we call this an *improving pivot operation*.

The modulo network simplex iteratively applies improving pivot operations to the current tree solution until it terminates with a solution, which cannot be improved further by exchanging a co-tree arc with a tree arc.

4 Integrating Passenger Routing

In order to integrate passenger routing into the modulo network simplex method, we replace the fixed arc weights w by a variable passenger routing along paths in the network N.

The passenger demand is given in terms of an *origin-destination matrix* (OD-matrix) $(d_{st}) \in \mathbb{Q}_{\ge 0}$ specifying for each pair $(s,t) \in V \times V$ the number of passengers that want to travel from s to t. Let $D = \{(s,t) \in V \times V : d_{st} > 0\}$ be the set of all *OD-pairs* and for an OD-pair (s,t) let $\mathcal{P}_{st}$ be the set of (s,t)-paths in N and $\mathcal{P} := \bigcup_{(s,t) \in D} \mathcal{P}_{st}$ be the set of all passenger paths.

We extend the model (PTT_w) to a version $(\mathrm{PTT}_\mathcal{P})$ with integrated passenger routing. We introduce passenger variables $f_p \ge 0$ for the fraction of passengers that travel on path $p \in \mathcal{P}$ and enforce the passenger flow by constraints $\sum_{p \in \mathcal{P}_{st}} f_p = 1$ for all $(s,t) \in D$. We include constraints (1)–(4) and change the objective as follows:

$$\min c(y,z,f) := \sum_{a \in A} \sum_{(s,t) \in D} \sum_{\substack{p \in \mathcal{P}_{st} \\ a \in p}} d_{st} f_p (y_a + \ell_a).$$

The resulting model $(\mathrm{PTT}_\mathcal{P})$ is a mixed-integer non-linear program that minimizes the total passenger travel time among all feasible timetables.

Theorem 2 *There exists an optimal solution* $(y^\mathcal{S}, z^\mathcal{S}, f^\mathcal{S})$ *of* $(\mathrm{PTT}_\mathcal{P})$ *such that* $(y^\mathcal{S}, z^\mathcal{S})$ *is a spanning tree solution, i.e., there exists a spanning tree structure* $\mathcal{S} = \mathcal{S}_\ell \cup \mathcal{S}_u$ *such that* $y^\mathcal{S}_a = 0$ *for all* $a \in \mathcal{S}_\ell$ *and* $y^\mathcal{S}_a = u_a - \ell_a$ *for all* $a \in \mathcal{S}_u$.

Proof Let (y^*, z^*, f^*) be an optimal solution of $(\mathrm{PTT}_\mathcal{P})$. Define arc weights $w^*_a := \sum_{(s,t) \in D} \sum_{p \in \mathcal{P}_{st} : a \in p} d_{st} f^*_p$, $a \in A$. Let $(y^\mathcal{S}, z^\mathcal{S})$ be an optimal spanning tree solution of (PTT_{w^*}) for the arc weights w^*. Since $(y^\mathcal{S}, z^\mathcal{S})$ is optimal and (y^*, z^*) is feasible for (PTT_{w^*}), we have:

$$c(y^{\mathrm{S}}, z^{\mathrm{S}}, f^*) = \sum_{a \in A} w_a^*(y_a^{\mathrm{S}} + \ell_a) \leq \sum_{a \in A} w_a^*(y_a^* + \ell_a) = c(y^*, z^*, f^*). \tag{5}$$

This inequality implies that $(y^{\mathrm{S}}, z^{\mathrm{S}}, f^*)$ is also an optimal solution of $(\mathrm{PTT}_{\mathcal{P}})$. □

Theorem 2 shows that it suffices to investigate spanning tree solutions as well when we integrate passenger variables. In the integrated case we have to consider the passenger flow in order to compute the difference in the objective value between two solutions. Let (y, z, f) be a feasible spanning tree structure solution of $(\mathrm{PTT}_{\mathcal{P}})$ and let y' be a feasible pivot operation. The passenger flow that minimizes the travel time with respect to the modified timetable y' is given by

$$f' := \operatorname{argmin}\left\{ c(y', z', f) : \sum_{p \in \mathcal{P}_{st}} f_p = 1 \, \forall (s,t) \in D, f \in [0,1]^{\mathcal{P}} \right\}.$$

Hence, y' is an improving pivot operation in the integrated case if $c(y', z', f') < c(y, z, f)$.

5 Computational Experiments

We implemented four variants of the modulo network simplex method in C++11 to assess the improvement potential of our integrated approach. We call the standard modulo network simplex method with fixed arc weights *static*. The variant with fully integrated passenger routing, which compares the objective values with updated passenger flows when searching for improving pivot operations, is called *integrated*. Since the integrated variant takes a toll on the runtime compared to the classic static variant, we also implemented an *iterative* version that applies the static modulo network simplex method and, at its end, updates the arc weights by passenger flow computations; this process is iterated until it cannot improve the solution any further. We finally tested a *hybrid* mode that updates the passenger flow induced arc weights after each pivot operation.

Instead of selecting the most improving pivot operation in each modulo network simplex iteration we used a faster "Quality First" rule as proposed in [3], which selects the first pivot with a satisfying improvement on the objective value. We used an improvement threshold of 0.1% and a scaling factor of 0.2 in all computations. A run was terminated after at most two hours plus finishing the incumbent iteration. All computations were done on an Intel(R) Xeon(R) CPU E3-1290 V2, 3.7 GHz computer (in 64 bit mode, 15 GB system memory), running Linux.

Statistics on four test instances are given in Table 1. The instance Wuppertal is based on the real multi-modal public transportation network of the city of Wuppertal for 2013. The remaining two Wuppertal-instances are obtained by selecting a subset of lines of this instance. The Dutch instance is based on a network that was introduced by Bussieck in the context of line planning. In all instances the lines are operated at different frequencies; their period times are 10, 15, 20, 30, or 60 min.

Table 1 The columns list the instances, the number of stations, the number of directed lines, the number of OD-pairs, the period time, the number of events, the number of activities, a lower bound on the optimal objective value for model ($\mathrm{PTT}_{\mathcal{P}}$), and the objective value of the starting solution

Instance	$\lvert S\rvert$	$\lvert\mathcal{L}\rvert$	$\lvert D\rvert$	T	$\lvert V\rvert$	$\lvert A\rvert$	Lower bound	Starting sol.
Wuppertal 98	123	98	32,857	20	1,370	10,994	2,043,083.52	2,239,330.56
Wuppertal 154	148	154	45,159	60	4,313	75,768	2,257,792.97	2,517,657.17
Wuppertal	1,582	311	196,158	60	13,202	78,090	5,016,813.33	5,625,657.98
Dutch	23	40	158	60	447	3,626	868,074.00	871,964.00

Table 2 Computational results. The columns list the instances, the variant of the algorithm, the computation time, the number of pivot iterations, the average time per pivot operation, the final objective value, the optimality gap compared to the lower bound, and the improvement compared to the starting solution. For the iterative method, the number of (outer) iterations is given in parentheses

Instance	Method	Time (s)	Pivot iter.	Time/iter.	Final obj.	Gap in %	Impr. in %
Wuppertal 98	Static	16	3	5.34	2,237,571.31	8.69	0.08
	Iterative (3)	25	6	4.18	2 233 814.57	8.54	0.25
	Integrated	7,232	48	150.66	2,161,064.73	5.46	3.50
	Hybrid	16	4	4.05	2,233,814.57	8.54	0.25
Wuppertal 154	Static	6,496	6	1,082.74	2,516,268.31	10.27	0.06
	Iterative (2)	7,115	7	1,016.41	2,515,421.58	10.24	0.09
	Integrated	7,479	14	534.21	2,457,124.12	8.11	2.40
	Hybrid	6,468	6	1,078.01	2,515,421.58	10.24	0.09
Wuppertal	Static	7,479	19	393.62	5,622,157.01	10.77	0.06
	Iterative (1)	7,490	19	394.20	5,622,157.01	10.77	0.06
	Integrated	8,206	7	1,172.34	5,553,853.73	9.67	1.28
	Hybrid	7,379	14	527.06	5,618,800.66	10.71	0.12
Dutch	Static	4	6	<1	871,697.00	0.42	0.03
	Iterative (2)	4	7	<1	871,697.00	0.42	0.03
	Integrated	147	18	8.18	868,320.00	0.03	0.42
	Hybrid	1	2	<1	871,772.00	0.42	0.02

Statistics on the computations are given in Table 2. The integrated variant apparently outperforms the others in terms of quality but at the cost of a strong increase in the computation time. The computations confirm the existence of substantial optimization potentials of integrating passenger routing into periodic timetable computations.

References

1. Borndörfer, R., Hoppmann, H., Karbstein, M.: Passenger routing for periodic timetable optimization. Public Transp. epub ahead of print (2016)
2. Gattermann, P., Großmann, P., Nachtigall, K., Schöbel, A.: Integrating Passengers' Routes in Periodic Timetabling: a SAT approach. In: Goerigk, M., Werneck, R. (eds.) ATMOS. Dagstuhl, Germany (2016)
3. Goerigk, M., Schöbel, A.: Improving the modulo simplex algorithm for large-scale periodic timetabling. Comput. Oper. Res. **40**(5) (2013)
4. Liebchen, C.: Periodic timetable optimization in public transport. Ph.D. thesis, Technische Universtität Berlin (2006)
5. Nachtigall, K.: Periodic network optimization and fixed interval timetables. Habilitation thesis, Universtität Hildesheim (1998)
6. Nachtigall, K., Opitz, J.: Solving periodic timetable optimisation problems by modulo simplex calculations. In: Fischetti, M., Widmayer, P. (eds.) ATMOS. Dagstuhl, Germany (2008)
7. Serafini, P., Ukovich, W.: A mathematical model for periodic scheduling problems. SIAM J. Discrete Math. **2**(4) (1989)

A Re-optimization Approach for Train Dispatching

Frank Fischer, Boris Grimm, Torsten Klug and Thomas Schlechte

Abstract The Train Dispatching Problem (TDP) is to schedule trains through a network in a cost optimal way. Due to disturbances during operation existing track allocations often have to be re-scheduled and integrated into the timetable. This has to be done in seconds and with minimal timetable changes to guarantee smooth and conflict free operation. We present an integrated modeling approach for the re-optimization task using Mixed Integer Programming. Finally, we provide computational results for scenarios provided by the INFORMS RAS Problem Soling Competition 2012.

1 Introduction

The Train Dispatching Problem (TDP) deals with the determination of a railway timetable by constructing train routes and corresponding arrival and departure times to operate train requests in a given railway network. Due to the complex operation rules, limited capacity, which is only upgradeable with massive financial effort, the infrastructure network builds a natural bottleneck. Thus, it is appreciable to utilize the existing infrastructure in the best way.

The TDP integrates several major requirements like safety system rules, train characteristics, blocking and headway times, timetable requirements, and infrastructure capacities. A detailed problem description and a Mixed Integer Programming formulation to solve this problem is described in detail in [5]. In this paper, we report

F. Fischer
Universität Kassel, Kassel, Germany
e-mail: frank.fischer@uni-kassel.de

B. Grimm · T. Klug (✉) · T. Schlechte
Zuse Institute Berlin, Takustr. 7, 14195 Berlin, Germany
e-mail: klug@zib.de

B. Grimm
e-mail: grimm@zib.de

T. Schlechte
e-mail: schlechte@zib.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_85

on a *Re-optimization* or *Re-scheduling* approach for the TDP in a real time setting using a state-of-the-art MIP solver.

The authors of [2] introduced a re-optimization approach for rolling stock rotation planning problems. In case of the TDP, we adapted it as follows: At some point in time a railway undertaking has to agree on a timetable, ideally, utilizing an optimization algorithm or by manual planning. Later in time this problem or aspects changes that much, such that the reference solution, in case of the TDP the timetable, becomes infeasible. Thus, a modified problem has to be solved. In contrast to the first process leading to the reference timetable the time limitations are in the second stage rather strict. Since an operator has only minutes or seconds for his decisions, the re-optimization algorithm has to calculate solutions within a real time management system. Another major goal is to change as few as possible in comparison to the original timetable. This should minimize the disturbance of the ongoing timetable because fewer changes are easier to communicate, easier to apply, and hence more reliable. Moreover, it is impossible for an operator to change many routes at the same time, because the running and blocking times highly depend on the routes and interaction between the trains. In case of the timetable construction this is evaluated in detail by microscopic simulation which is not applicable in a real time setting. Therefore, the reference solution highly influences the objective function. There are various causes that can lead to a situation where the implemented timetable becomes unexpectedly infeasible. Predictable and unpredictable construction sites and breakdowns that block a track must be integrated into the timetable as fast as possible. In addition, delayed trains and modifications of speed limits may require an adjustment of the timetable. The paper contributes an adaption of the Mixed Integer Programming approaches presented in [5, 7] to re-optimize timetables. We show how to incorporate re-optimization requirements into the disjunctive graph based formulation, see [1, 3, 4, 6]. An iterative approach is used by [4] to solve real-time instances of the Dutch railway network. They use a branch-and-bound algorithm for sequencing train movements and improving the solution by locally rerouting some trains. The connection between adjacent dispatching areas is studied by [3]. Mascis and Pacciarelli [6] use a disjunctive graph formulation to model and solve a job-shop scheduling problem with blocking constraints. This paper is organized as follows. Section 2 defines the considered problem including an overview of the disjunctive based formulation. In Sect. 3 we present some real world scenarios, consider common re-optimization use cases for the TDP, and presents computational results. This indicates that the model and algorithmic approach produces high quality solutions in a very short time and is able to tackle the real time re-optimization setting.

2 A Re-optimization Model for the TDP

Consider the following problem setting for the TDP. We model the infrastructure network by a directed graph $G = (V, A)$. The arcs correspond to track segments with fixed running times $\tau^r_{(v,w)}$ for each train r that is able to operate on track segment

$(v, w) \in A$. For each track segment (v, w) and train pair r, r' exists a headway time $h^{r,r'}_{(v,w)}$ which is defined as the minimal time between two consecutive trains r and r' that use the same track segment (v, w), see details in [9]. The set of scheduled train requests is denoted by R. Each train $r \in R$ is associated with an initial route $p^* \in P^r$, where P^r is the set of possible routes for request r. Additionally, there are essential stops $S^r \subset V$ for each train $r \in R$. Each stop $s \in S^r$ of train r have to be fulfilled during the time period $[\underline{\alpha}^r_s, \overline{\alpha}^r_s]$. A time unit of deviance from the scheduled departure of train r, denoted by α^r_s, is penalized by $\underline{c}^r_s$ if the actual departure time is before α^r_s, and $\overline{c}^r_s$ if the actual departure time is after α^r_s. Furthermore, we denote by δ^r_p the cost that occur from re-routing train r on route p instead of its initial route p^* with cost $\delta^r_{p^*} = 0$. By $\gamma^r \in \mathbb{R}^-$ we denote a (negative) profit value for not routing train r. Usually, this value should ensure that a maximal number of trains is scheduled. If meaningful data is available this could also be used to give the algorithm a priority estimation for each train depending on the demand. The set $B \subseteq A$ is the set of arcs (v, w) where some kind of disturbance takes place during the period of $[\underline{\beta}_{(v,w)}, \overline{\beta}_{(v,w)}]$.

A solution of the TDP has to associate each scheduled train $r \in R$ at most one route $p \in P^r$ with departure times for each node $v \in p$ under consideration of the headway constraints. The task of the model is to select a path for each train and to determine departure times t^r_v for each node v that is visited by train r on its path. For this, we enforce relations between different departure times w. r. t. the chosen paths and the order in which different trains traverse the same arc. In particular, we will make use of the following three types of decisions:

1. r uses (v, w), which is satisfied if and only if the selected path for r contains arc (v, w),
2. $r \prec_{(v,w)} r'$, which is satisfied if and only if r traverses (v, w) before r',
3. $r \prec b_{(v,w)}$ and $r \succ b_{(v,w)}$, which are satisfied if and only if r uses (v, w) before or after the disruption, respectively.

Depending on these conditions, we can formulate the following constraints for the departure times:

$$\textit{running times:} r \text{ uses } (v, w) \qquad \Rightarrow t^r_v + \tau^r_{(v,w)} \leq t^r_w, \tag{1}$$

$$\textit{headway times:} r \prec_{(v,w)} r' \qquad \Rightarrow t^r_v + h^{r,r'}_{(v,w)} \leq t^{r'}_v, \tag{2}$$

$$\textit{disruption times:} r \prec b_{(v,w)} \quad \Rightarrow t^r_w \leq \underline{\beta}_{(v,w)}, \text{ and } r \succ b_{(v,w)} \Rightarrow t^r_v \geq \overline{\beta}_{(v,w)}. \tag{3}$$

We using the following binary variables

1. $z^r_p = 1 \iff r$ runs on $p \in P^r$,
2. $x^{r,r'}_{(v,w)} = 1 \iff r$ runs before r' on (v, w),

3. $b^r_{(v,w)} = 1 \iff r$ runs before disruption on (v, w)

and formulate the disjunctive constraints as linear big-M constraints.

With this notation the TDP can be stated as a mixed integer program as follows:

$$\text{minimize} \quad \sum_{v \in S} (\underline{c}^r_v \underline{\Delta}^r_v + \overline{c}^r_v \overline{\Delta}^r_v) + \sum_{r \in R} \sum_{p \in P^r} \delta^r_p z^r_p + \sum_{r \in R} \gamma^r \sum_{p \in P^r} z^r_p \tag{4}$$

$$\text{subject to} \quad (1), (2), (3), \tag{5}$$

$$\sum_{p \in P^r} z^r_p \leq 1, \qquad r \in R, \tag{6}$$

$$t^r_v - \overline{\Delta}^r_v \leq \alpha^r_v, \qquad r \in R, v \in S^r, \tag{7}$$

$$t^r_v + \underline{\Delta}^r_v \geq \alpha^r_v, \qquad r \in R, v \in S^r, \tag{8}$$

$$t^r_v \in [\underline{\alpha}^r_v, \overline{\alpha}^r_v], \qquad r \in R, v \in S^r, \tag{9}$$

$$t^r_v, \underline{\Delta}^r_v, \overline{\Delta}^r_v \geq 0, \qquad r \in R, v \in S^r, \tag{10}$$

$$z, x, b \text{ binary} \tag{11}$$

In addition to the three types of binary variables, the continuous variables t^r_w model the departure time of train r at node w. The continuous cost variables $\underline{\Delta}^r_w$ and $\overline{\Delta}^r_w$ measure the deviation between the departure time of the reference timetable and re-allocated departures times of train r at node w. The linear objective function (4) minimizes the sum of the total costs for deviance at stops, costs for alternative routes, and costs for unscheduled trains. The constraints for the running times (1), the headway times (2) and the disruption times (3) are formulated as big-M constraints as mentioned above. The inequalities (7) and (8) ensure the correct values for the time deviation cost variables and constraints (6) ensure that at most one route is selected for each train. The departure time windows of the stops are modeled by (9).

If the trains have delays the model aims at pushing the trains back to its actual routing and timing. In some cases this is not desired since the new schedule may lead to a lot of modifications of the current timetable, which is not realizable. In this case the reference departure times could be adjusted to keep the delays at the current level. Of course, by setting the variables cost $\underline{c}^r_s$, $\overline{c}^r_s$ and δ^r_p to zero for all stops, paths and trains it is possible to calculate a timetable that is completely independent from the reference timetable.

Table 1 Key numbers of re-optimization scenarios from RAS

Instance	$\|R\|$	Disrupted arcs	Disrupted routes	Planning horizon (h:m:s)
RAS_1	12	2	5(41%)	17:58:47
RAS_2	18	2	6(33%)	18:07:47
RAS_3	20	12	18(90%)	16:32:15

3 Computational Study

We implemented the proposed re-optimization model in a C++-framework. This implementation takes use of MIP solver CPLEX 12.6. All our computations were performed on a desktop computer with an Intel Xeon CPU E3-1245 v3 with 3.40 GHz and 32 GB of RAM. The set of instances are scenarios derived from the INFORMS RAS Problem Solving Competition 2012, see [8].

The *RAS* instances include a microscopic infrastructure network containing 82 nodes and 184 arcs. There are three different scenarios with increasing complexity, i.e., in terms of larger number of trains and disturbances. Table 1 shows the corresponding sizes.

In all cases we chose $\gamma^r = -10^3$ for the profit value of routing train r. The parameter δ_p^r equals the number of deviating tracks between route p and reference route p^*. The cost parameters are set in such a way that the optimization goals are weighted in order of importance. First the number of cancelled trains should be minimal, second the number of route changes should be minimal and third the departure times should be as close as possible at the reference timetable. For the MIP solvers we set a time limit of 1800 s.

We limited the set of possible routes for each train since otherwise most of the trains have 192 possible routes which is far too much to handle. In addition, most of those ignored potential routes cannot be part of an optimal solution. An observation is that the model can be solved in a few seconds if the number of alternative routes per train is small. Therefore we sort accordingly to δ_p^r the alternative routes for each train and select the first 4, 8 and 16 alternative routes for each train, respectively. We use the cost parameters δ_p^r since there are the only costs that can be calculated in advance.

The computational results are in Table 2. The second column is the number of alternative routes for each train and is followed by the number of trains in the reference time table. Then we have the number of blocked trains and the number of planned, cancelled and rerouted trains in the solution. If the time limit was reached than this is indicated with TL in the running time column.

From the practical point of view even the restriction to four tours per train is more than a dispatcher can overlook in a couple of minutes or even seconds. We are able to solve the first two scenarios to optimality and solve the third with an optimality gap of at most 5.3%. It turns out that for the RAS instances the first four selected tours

Table 2 Solutions of the RAS scenarios with 4, 8 and 16 alternative routes for each train

Instance	Alt. routes	Trains	Blocked	Planned	Cancelled	Route changes	Running time (s)	Gap (%)	Gap after 20 s (%)	Objective
RAS_1	4	12	0	12	0	0	3.0	0.0	0.0	−11852.72
RAS_1	8	12	0	12	0	0	17.0	0.0	0.0	−11852.72
RAS_1	16	12	0	12	0	1	51.0	0.0	21.3	−11993.00
RAS_2	4	18	0	18	0	2	12.0	0.0	0.0	−17869.33
RAS_2	8	18	0	18	0	2	89.0	0.0	>100.0	−17869.33
RAS_2	16	18	0	18	0	2	1492.0	0.0	>100.0	−17871.33
RAS_3	4	20	0	19	1	1	982.0	0.0	5.3	−18731.33
RAS_3	8	20	0	19	1	1	TL	5.3	11.4	−18731.33
RAS_3	16	20	0	19	1	4	TL	5.4	40.1	−18713.33

are sufficient to provide high quality solutions. Furthermore the optimality gaps after 20 s indicate that we are able to get good solutions fast.

4 Conclusion

We extended a well known MIP formulation for the TDP to be able to tackle re-optimization scenarios. Our computational study demonstrates that our re-optimization approach can be used to produce high quality solutions in reasonable computation time for a real time application.

References

1. Balas, E.: Machine sequencing via disjunctive graphs. Oper. Res. **17**, 941–957 (1969)
2. Borndörfer, R., Mehrgardt, J., Reuther, M., Schlechte, T., Waas, K.: Re-optimization of rolling stock rotations. Technical Report 13–60, ZIB, Takustr.7, 14195 Berlin (2013)
3. Corman, F., D'Ariano, A., Pacciarelli, D., Pranzo, M.: A bilevel rescheduling framework for optimal inter-area train coordination. In: ATMOS, pp. 15–26 (2011)
4. D'Ariano, A., Corman, F., Pacciarelli, D., Pranzo, M.: Reordering and local rerouting strategies to manage train traffic in real time. Transp. Sci. **42**(4), 405–419 (2008)
5. Mannino, C.: Real-time traffic control in railway systems. In Alberto, C., Spyros, K. (eds.) ATMOS, vol. 20, Dagstuhl, Germany (2011) Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik
6. Mascis, A., Pacciarelli, D.: Job-shop scheduling with blocking and no-wait constraints. Eur. J. Oper. Res. **143**(3), 498–517 (2002)
7. Pellegrini, P., Marlière, G., Rodriguez, J.: Real time railway traffic management modeling track-circuits. In: ATMOS, volume 25 of OpenAccess Series in Informatics (OASIcs), Dagstuhl, Germany, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, pp. 23–34 (2012)
8. INFORMS Railway Application Section (RAS). Problem-solving-competition 2012. https://www.informs.org/Community/RAS/Problem-Solving-Competition/2012-RAS-Problem-Solving-Competition (2012)
9. Schlechte, T.: Railway track allocation: models and algorithms. Ph.D. thesis, TU, Berlin (2012)

Electric Vehicle Scheduling—A Study on Charging Modeling for Electric Vehicles

Nils Olsen and Natalia Kliewer

Abstract Electric Vehicle Scheduling extends the traditional Vehicle Scheduling by restricting the range of deployed vehicles and considering the possibility to recharge a vehicle at some charging stations. A fundamental aspect of electro-mobility which hasn't attract much attention within existing solution approaches for the E-VSP, is the functionality of charging an electric vehicle's battery. In this paper, we propose models for the charging process of electric vehicles and analyse their impacts on resulting vehicle schedules. We focus on the question whether widely used assumptions concerning the charging times of electric vehicles, for instance constant or linear charging times, reflect the reality of electro-mobility sufficiently or should be considered in a more accurate way.

1 Introduction

Scheduling a fleet of vehicles represents a fundamental task within the planning process of companies in public transportation. The mathematical optimization problem which results from this task is widely known as the *Vehicle Scheduling Problem* (VSP), which determines the vehicle deployment for serving timetabled service trips. The VSP is a well studied problem in the research community and has been widely analyzed (cf. [5]).

In the course of global warming and the proceeding rejection of fossil energy sources towards renewable energies, the importance of alternative engines in the transport sector has increased strongly. Electro-mobility occupies a special position in the scope of alternative engines since electric vehicles (EVs) enable an emission-free movement. Their crucial properties are the much shorter range compared to combustion engine vehicles, due to the restricted battery capacity and the possibility to recharge their batteries at charging stations (cf. [13]). Despite of profound research

N. Olsen (✉) · N. Kliewer
Freie Universität Berlin, Garystr. 21, 14195 Berlin, Germany
e-mail: nils.olsen@fu-berlin.de

N. Kliewer
e-mail: natalia.kliewer@fu-berlin.de

A. Fink et al. (eds.), *Operations Research Proceedings 2016*,
Operations Research Proceedings, DOI 10.1007/978-3-319-55702-1_86

in the area of battery technologies, modern EVs merely reach a fractional part of the range of conventional vehicles (cf. [10]). For companies in public transportation, there arise new challenges for their planning processes, especially within vehicle scheduling, when providing their services with EVs. Due to the electro-mobility restrictions, solution methods for the traditional VSP cannot be applied to the electric case. If we ignore this aspect EVs would very likely stop within their rotations when they run out of energy and cause high costs, due to recovery services or dissatisfied passengers. The resulting mathematical optimization problem is denoted as the *Electric Vehicle Scheduling Problem* (E-VSP), which is a very current research topic. In the last few years, researchers have focused on developing solution approaches for the E-VSP. As a first approach, [7, 8] extended the traditional VSP by restricting the length of the vehicles' paths, but neglect the possibility to recharge a vehicle's battery. [6] consider, beside a limited range, the possibility to swap a vehicle's battery, which can be considered as a constant charging time. [1] present a column-generation approach, which incorporates both a limited range and chargings at fixed locations, which is also done in constant time. For generating initial solutions, an heuristic algorithm is proposed, which generates vehicle schedules greedily with respect to electro-mobility constraints. [12] develop a column-generation approach, which considers partial chargings of EVs by a discrete set of states.

Within existing solution approaches for the E-VSP, there are some fundamental aspects of electro-mobility which have remained unconsidered up to now. In this context, determination of the underlying charging infrastructure or more accurate reflections of an EV's technical properties need to be considered. Technical issues concern battery life-cycles, battery degeneration or the charging and discharging process of electric batteries. A more realistic consideration of these aspects may likely lead to optimization potentials, due to the more realistic mapping and increasing degrees of freedom.

In this paper we focus on the charging process of EVs. We face the question whether simplified assumptions, like constant or linear charging times as used in [1] or [12], can be justified in practice, or whether a more specific consideration is necessary. In practice, most EVs use lithium ion batteries to power their engines (cf. [6] or [13]). [3] present a crucial property of lithium ion batteries, concerning the required time to recharge a battery depending on its residual energy. Accordingly, for charging a battery from zero to approximately 65% of its battery capacity, the actual percentage value depends on the C-rate of the battery, the maximum charging ratio, the amount of energy which can be fed into a battery per time unit, is available while the charging ratio decreases quickly when exceeding this threshold. As a consequence, a lithium ion battery can be charged to an amount of energy less or equal to 65% in a relatively short time frame, whereas the time needed to fully charge the battery may take a multiple of that time. This coherence between the charging time and the residual energy is often connected to the most utilized charging mode *constant current/constant voltage* (CC/CV) (cf. [9]). In general, the charging process of an EV follows a nonlinear pattern depending on its residual energy, the charging system or other circumstances. Thus, a charging process comprises two components which are mutually linked: a specific charging time during a vehicle stops at a charg-

ing station and a specific amount of energy, which is fed into the vehicle's battery within this time frame.

In order to answer the introduced question the paper is organized as follows: in Sect. 2 we define the E-VSP and introduce a heuristic solution method for the E-VSP from [1], which we will use later. In Sect. 3 we present a modeling for the charging process of EVs and perform a computational study in Sect. 4 to analyse the results and give key statements.

2 The Electric Vehicle Scheduling Problem

The objective of the traditional VSP is to determine a cost-optimal assignment of a set of timetabled service trips to a set of vehicle rotations while satisfying the following constraints: (1) A vehicle's schedule begins and ends at the same depot, (2) trips of a vehicle's schedule are mutually compatible and (3) every service trip is covered exactly once. Then, each vehicle rotation represents a sequence of trips, which a vehicle executes consecutively. The E-VSP extends the VSP by additional constraints due to the deployment of EVs: (4) A vehicle's residual energy cannot fall below zero and cannot exceed its battery capacity and (5) a vehicle can only be recharged at stoppoints, which are equipped with charging technology.

To evaluate the modeling approaches for the charging process of EVs, we use a heuristic solution method from [1] based on a concurrent greedy algorithm. The basic procedure is to consecutively assign service trips to the set of already used vehicles, subject to electro-mobility constraints. If the range restriction is violated a charging process is inserted if there's enough time for charging. From the set of used vehicles able to execute the current service trip, the one is chosen which causes minimum additional operative costs. Is there no such vehicle a new one is added. The procedure is repeated until every service trip is covered.

3 Modeling Approaches for the Charging Process of EVs

As stated previously, the charging process of EVs follows a nonlinear pattern, which is influenced by several factors. To incorporate external influences beside a vehicle's residual energy, we assume that the charging process of an EV is given by a function

$$e\left(x_1, \ldots, x_n, c\right) : X_1 \times \cdots \times X_n \times [0, c_{max}] \to \mathbb{R}_{\geq 0} \tag{1}$$

which indicates the amount of energy (measured in kWh) that can be fed into the battery per minute depending on countably many influencing factors X_i, as well as the residual energy $c \in [0, c_{max}]$ whereby $c_{max} > 0$ represents the battery capacity. For reasons of simplification, we concentrate on a vehicle's residual energy and neglect any other factors, i.e. we assume $X_1, \ldots, X_n = \emptyset$. (1) is denoted as the *charging ratio*.

If a vehicle arrives at a charging station with residual energy c, the required charging time $z \in \mathbb{Z}$ in min for charging its battery to a charge $\beta \in [c, c_{max}]$ is given implicitly by

$$\beta = c + \int_{c}^{\alpha} e(x)\, dx \tag{2}$$

with $\alpha \geq c$ and $z = \lceil \alpha - c \rceil$. Depending on the shape of e, the charging time z may be computed analytically or has to be approximated, if the integral is not computable or doesn't exist. In these cases we use Newton-Cotes formulas together with Newton's method to approximate z (cf. [11]). In order to find appropriate shapes of (1), we gradually approach the nonlinear pattern described in Sect. 1. To enable a comparison between constant resp. linear charging times and more complex ones we assume a linear charging ratio

$$e(c) = a \cdot c + b \tag{3}$$

with $a < 0$ and $b > 0$, first. (3) is strictly monotonically decreasing and depends strongly on the choice of a and b. Thus, (3) implies a decreasing charging ratio relative to the residual energy, as it was stated before. The parameters a and b must be chosen in such a way that (3) always remains positive. As a representative of nonlinear charging ratios we propose a logarithmical charging ratio in the form of

$$e(c) = \begin{cases} a \cdot log(c) + b\ , & c \geq lb \\ b\ , & else \end{cases} \tag{4}$$

with $a, b \in \mathbb{R}$ and a lower bound $lb \in [0, c_{max}]$. The case differentiation is used to avoid negative charging ratios and enable more accurate modeling.

4 Computational Study

A charging process contains a holding time at a charging station and an amount of energy, which is fed into a vehicle's battery during this period. A constant charging time implies that residual energies are completely neglected, whereas a linear charging time reflects residuals in a linear way. In order to evaluate these model assumptions with regard to a battery's internal processes, we use them for determining required charging times but, however, we use the more realistic models of the charging ratio to determine resulting amounts of energy that are fed into a battery within the respective charging time. For this purpose we assume a constant charging time of $t = 30$ min and fit the generic charging ratios (3) and (4) in such a way that a full charging takes particularly t minutes when a battery is completely empty. t can be seen as a comparative value but any other choice may be reasonable too. Since (3) and (4) don't reflect the hard change of the charging ratio with regard to CC/CV we use an interpolation of the charging ratio proposed in [3], additionally.

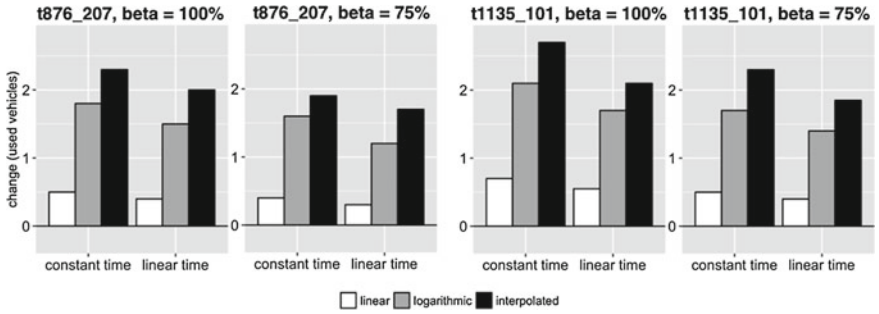


Fig. 1 Average changes in number of used vehicles using constant and linear charging time and linear, logarithmical and interpolated charging ratio

We solve two instances of the E-VSP by the algorithm described in Sect. 2, which differ in the number of service trips (876 vs. 1135) and number of stoppoints (207 vs. 101). The instances' names contain these two aspects. The charging process within the algorithm follows the procedure described in Sect. 3. We analyse the number of used vehicles, since this has the most significant impact on the total costs and helps us to answer the introduced question. Both instances are based on real-world data of German public transport companies but are modified to address electro-mobility. Within the respective route network, 5% of all stoppoints are equipped with charging technology and are sampled 20 times. Thus, the following results comprise average values. Due to nonlinear shapes of charging ratios the bounds until a battery is charged plays an important role when computing charging times. For that reason, we distinguish between a full charging ($\beta = 100\%$) and a charging up to $\beta = 75\%$.

Figure 1 illustrates the average changes in the number of deployed vehicles using the constant resp. linear time frames for charging together with each of the proposed charging ratios. We can observe that in both instances, as well as in both scenarios of β, the computed numbers of used vehicles exceed the original numbers when using more complex charging ratios. This is reasonable, because the constant and linear time frames for charging are not sufficient when the linear, logarithmical and interpolated charging ratios are assumed as the actual ones. Due to the monotonically decreasing profiles the time needed to fully recharge a vehicle exceeds the constant and linear time frame. Consequently, vehicles continue their rotations too early and, thus, charged amounts of energy are lower than actually required, which leads to an increase of used vehicles. Furthermore, the nearer we approach the interpolated charging ratio, which likely represents a realistic modeling, the more vehicles are used. In addition, we can observe that the numbers of used vehicles for $\beta = 75\%$ are lower than for $\beta = 100\%$. This is comprehensible because computed charging times by the linear, logarithmical and interpolated approaches come closer to the constant and linear time frames when charging up to 75% of a vehicle's battery capacity.

5 Summary and Conclusion

In this paper, we focused on the charging process of EVs within solution methods for the E-VSP. In both instances and both scenarios we revealed major gaps between model assumptions concerning an EV's charging time and actually loaded energy. This inconsistency leads to shorter vehicle rotations between charging stations as actually computed and to an increasing number of deployed vehicles. Furthermore, the bound of energy until a vehicle's battery is charged directly influences resulting vehicle schedules especially when more accurate modeling of the charging ratio is considered. This crucial aspect of an EV's charging process may be incorporated in further research.

References

1. Adler, J.D., Mirchandani, P.B.: The vehicle scheduling problem for fleets with alternative-fuel vehicles. In: Transportation Science, Articles in Advance, INFORMS, pp. 1–16 (2016)
2. Ball, M.: A comparison of relaxations and heuristics for certain crew and vehicle scheduling problems. In: ORSA/TIMS Meeting, Washington, D.C. (1980)
3. Battery University: BU-409: Charging Lithium-ion. http://batteryuniversity.com/learn/article/charging_lithium_ion_batteries (2016)
4. Botsford, C., Szczepanek, A.: Fast charging vs. slow charging: pros and cons for the new age of electric vehicles. In: EVS24 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, Stavanger, Norway, pp. 1–9 (2009)
5. Bunte, S., Kliewer, N.: An overview on vehicle scheduling models. Public Transp. **1**, 299–317 (2009)
6. Chao, Z., Xiaohong C.: Optimizing battery electric bus transit vehicle scheduling with battery exchanging: model and case study. In: Intelligent and Integrated Sustainable Multimodal Transportation Systems, 13th ITCP, Procedia—Social and Behavioral Sciences, vol. 96, pp. 2725–2736 (2013)
7. Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F.: Time constrained routing and scheduling. In: Monma, C.L., Nemhauser, G.L. (eds.) Network Routing, Handbooks in Operations Research and Management Science, vol. 8, pp. 35–139, North-Holland, Amsterdam (1995)
8. Haghani, A., Banihashemi, M.: Heuristic approaches for solving large-scale bus transit vehicle scheduling problem with route time constraints. Transp. Res. Part A **36**(4), 309–333 (2002)
9. Maxim Integrated: Application Node 4196: Understanding Li+ Battery Operation Lessens Charging Safety Concerns. http://pdfserv.maximintegrated.com/en/an/AN4169.pdf (2008)
10. Ogden, J.M., Steinbugler, M.M., Kreutz, T.G.: A comparison of hydrogen, methanol and gasoline as fuels for fuel cell vehicles: implications for vehicle design and infrastructure development. J. Power Sour. **79**(2), 143–168 (1999)
11. Schwarz, H.R., Kckler, N.: Numerische Mathematik. In: Teubner, Stuttgart 2006, 6. Auflage, pp. 311–316 (2006). ISBN 3-519-42960-8
12. van Kooten Niekerk, M.E., van den Akker, J.M., Hoogeveen, J.A.: Scheduling electric vehicles. In: Technical Report UU-CS-2015-013, Department of Information and Computing Sciences Utrecht University, Utrecht, The Netherlands (2015)
13. Wang, M., Zhang, R., Shen, X.S.: Mobile Electric Vehicles Online Charging and Discharging. Springer International Publishing (2016)