Operations Research Proceedings

Bernard Fortz
Martine Labbé *Editors*

Operations Research Proceedings 2018

Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Brussels, Belgium, September 12-14, 2018

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

Bernard Fortz • Martine Labbé
Editors

Operations Research Proceedings 2018

Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), Brussels, Belgium, September 12-14, 2018

Springer

Editors
Bernard Fortz
Department of Computer Science
Université Libre de Bruxelles
Brussels, Belgium

Martine Labbé
Department of Computer Science
Université Libre de Bruxelles
Brussels, Belgium

ISSN 0721-5924 ISSN 2197-9294 (electronic)
Operations Research Proceedings
ISBN 978-3-030-18499-5 ISBN 978-3-030-18500-8 (eBook)
https://doi.org/10.1007/978-3-030-18500-8

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

Preface

This book contains a selection of refereed short papers presented during the 2018 edition of the annual international conference of the German Operations Research Society e.V. (GOR), co-organized this year with the Belgian Operational Research Society (ORBEL). The conference took place in Brussels, Belgium, September 12–14, 2018, at the MCE Conference Centre.

With over 400 presentations covering the major current trends in Business Analytics, Artificial Intelligence and Forecasting, Control Theory and Continuous Optimization, Decision Theory and Multiple Criteria Decision Making, Discrete and Integer Optimization, Energy and Environment, Finance, Game Theory and Experimental Economics, Graphs and Networks, Health Care Management, Logistics and Freight Transportation, Metaheuristics, Optimization under Uncertainty, OR in Engineering, Pricing and Revenue Management, Production and Operations Management, Project Management and Scheduling, Simulation and Statistical Modeling, Software Applications and Modeling Systems, Supply Chain Management, and Traffic, Mobility and Passenger Transportation, the program gave the participants a great opportunity to discover what Operations Research has to offer today and where it is headed to.

Altogether, 70 submissions have been accepted for this volume (acceptance rate 67%), including 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 stream chairs, the 10 invited plenary and semi-plenary speakers, our exhibitors and sponsors, the very efficient team at MCE, the many people organizing the conference behind the scenes, and, last but not least, the participants from 36 countries. We hope that you enjoyed the conference as much as we did.

Brussels, Belgium
January 2019

Bernard Fortz
Martine Labbé

Contents

Part I
Awards

On a Polynomially Solvable Subclass of the Clique Problem with Applications in Energy-Efficient Timetabling

Patrick Gemander

1 Introduction

In many applications, especially scheduling applications, there are multiple decisions with a set of mutually exclusive options for each decision and possible conflicts between some of these options. This can be viewed as a clique problem with additional structure. In the *Clique Problem under Multiple-Choice Constraints* (CPMC) we are given a graph $G = (V, E)$ together with a partition $\mathcal{V} = \{V_1, \ldots, V_m\}$ of V such that all subsets V_i are stable sets in G. We then want to compute a clique C of size m, i.e. we choose exactly one vertex from each subset V_i such that the chosen vertices induce a complete subgraph of G.

In the context of CPMC, a decision is represented by a subset V_i and its options by the elements $v \in V_i$. Compatible options are connected by edges in E. For example, a scheduling problem can be viewed as the task to choose from departure times $v \in V_t$ for each train t such that all chosen departure times are compatible to each other, i.e. the overall schedule is valid. In our application, we choose from slightly altered departure times and speed profiles in order to increase the energy efficiency of an already existing underground or railway schedule. This is an extension of the work done in [3] and [7]. Another possible application for the present results could be runway scheduling as in [1].

CPMC has also been studied in [2], where the authors identify a first polynomial-time solvable special case referred to as *staircase compatibility*. In that case, certain conditions are imposed on the structure of the subgraphs induced by the union $V_i \cup V_j$ of any two distinct subsets V_i and V_j.

P. Gemander (✉)
FAU Erlangen-Nürnberg, Erlangen, Germany
e-mail: patrick.gemander@fau.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_1

In this research work, we show that CPMC is $\mathcal{NP}$-complete in the general case and identify a new special case which is also solvable in polynomial time. It does not rely on the structure of the subgraph induced by two subsets, but restricts which pairs of subsets in $\mathcal{V}$ may have mutually incompatible elements. We will be able to show that CPMC is solvable in polynomial time if the so-called "dependency graph" is a forest. This result stems from our finding that in this case the graph G is perfect. Furthermore, to describe the convex hull of the set of feasible solutions it is in this case sufficient to consider only the stable-set inequalities induced by subgraphs on $V_i \cup V_j$. In other words, the separation problem reduces to a maximum-weight stable-set problem on a bipartite graph. This property will make it possible to derive non-trivial upper bounds on the number of stable-set inequalities needed to describe the convex hull. Moreover, it enables us to use a network flow algorithm to solve the corresponding separation problem in polynomial time. In a computational case study, we will show that there is significant benefit from exploiting these properties.

2 Problem Definition

As CPMC is a special case of the general clique problem, we first proved CPMC to be $\mathcal{NP}$-complete via a reduction from the satisfiability problem (SAT) [6] and later discovered an even easier reduction from the colouring problem [4].

Theorem 1 *The decision variant of CPMC is $\mathcal{NP}$-complete.*

An intuitive formulation of CPMC as a binary programming problem is given as follows:

$$\begin{aligned}
&\text{find} && x && && \\
&\text{s.t.} && \sum_{v \in V_i} x_v = 1 && \forall V_i \in \mathcal{V} && (1)\\
& && x_v \leq \sum_{\substack{u \in V_j \\ \{u,v\} \in E}} x_u && \forall V_i, V_j \in \mathcal{V}, i < j, \forall v \in V_i && (2)\\
& && x_v \in \{0,1\} && \forall v \in V. && (3)
\end{aligned}$$

To properly identify the polynomial-time solvable special case we need the following definitions.

Definition 1 (Subgraphs G_{ij}, Dependency Graph, FCPMC) Let $\mathcal{I} = (G, \mathcal{V})$ with $G = (V, E)$, $\mathcal{V} = \{V_1, \ldots, V_m\}$ be an instance of CPMC.

For $V_i, V_j \in \mathcal{V}, i \neq j$ we write

$$G_{ij} = (V_i \cup V_j, E_{ij})$$

for the subgraph of G induced by $V_i \cup V_j$. Further, we call $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with

$$\mathcal{E} = \{\{V_i, V_j\} \mid i \neq j \wedge \exists u \in V_i, v \in V_j : \{u, v\} \notin E\}$$

the *dependency graph* of G.

Let $\mathcal{I}$ be an instance of CPMC such that the corresponding dependency graph $\mathcal{G}$ is a forest, i.e. does not contain cycles. We then call $\mathcal{I}$ an instance of *FCPMC*.

The name 'dependency graph' stems from the fact that for choosing a clique in G, the choice of a vertex in V_i has immediate consequences for the choice of a vertex in V_j if the edge $\{V_i, V_j\}$ is contained in $\mathcal{G}$.

3 The FCPMC Polytope

We begin by stating our main result. For FCPMC, we are able to give the complete convex-hull description.

Theorem 2 *Let $\mathcal{I} = (G, \mathcal{V})$ be an instance of FCPMC. Then the FCPMC polytope $P_{FCPMC}(\mathcal{I}) = conv(\{x \in \{0,1\}^{|V|} \mid x$ incidence vector of a solution of $\mathcal{I}\})$ is determined by*

$$\sum_{v \in V_i} x_v = 1 \qquad \forall V_i \in \mathcal{V} \tag{4}$$

$$\sum_{v \in S} x_v \leq 1 \qquad \forall \text{ stable sets } S \subseteq V \tag{5}$$

$$x_v \geq 0 \qquad \forall v \in V. \tag{6}$$

In the proof, we show that G contains neither odd holes nor odd antiholes of length greater than or equal to five. Therefore the graph G is perfect by the Strong Perfect Graph Theorem [5]. In the last step, we transform the convex-hull description of the general clique problem on G into a convex-hull description of CPMC on G. The complete proof can be found in [6] or [4]. This result immediately yields the following corollary.

Corollary 1 *FCPMC is in $\mathcal{P}$.*

Unfortunately, adding the full convex-hull description is not a feasible approach in the general case.

Theorem 3 *There exists a family of instances such that the corresponding FCPMC polytopes have a number of facets exponential in the number of vertices in their respective graphs.*

A precise description of this family of instances as well as the proof to Theorem 3 can be found in [6] or [4].

Although the FCPMC polytope has exponentially many facets in the general case, the following results allowed us to add the entire convex-hull description to our MIP models for solving underground and railway scheduling problems. First, Lemma 1 greatly reduces the search space concerning stable sets.

Lemma 1 *Let $S \subseteq V$ be a stable set in G. Then there exists a subgraph G_{ij} of G such that $S \subseteq V_i \cup V_j$.*

Furthermore, so-called "non-dominating sets" allowed us to state a non-trivial upper bound on the number of facets.

Definition 2 Let G_{ij} be a subgraph of G as in Definition 1, $U \subseteq V_i$ and $N_{ij}(v) = \{w \in V_i \cup V_j \mid \{v, w\} \in E\}$. The subset U is a *non-dominating set* if for any vertex $u \in U$ we have

$$N(u) \not\subseteq \bigcup\nolimits_{v \in U \setminus \{u\}} N(v).$$

Theorem 4 *For each subgraph $G_{ij}, i < j$, let n_i^{ij}, n_j^{ij} be upper bounds on the size of non-dominating subsets of V_i and V_j with respect to G_{ij}. Then the number of facets of the FCPMC polytope $P_{FCPMC}(\mathcal{I})$ induced by stable-set inequalities is bounded from above by*

$$\sum\nolimits_{G_{ij}, i<j} \min \left\{ \sum\nolimits_{k=1}^{n_i^{ij}} \binom{|V_i|}{k}, \sum\nolimits_{k=1}^{n_j^{ij}} \binom{|V_j|}{k} \right\}.$$

Although the upper bound is exponential in the general case, we discovered that in our application it is at most cubic and even linear for large portions of the problem. This allowed us to determine and add the complete convex hull to the model and solve the problem efficiently. If that were not the case, one could separate stable-set inequalities by solving min-cut problems on the bipartite graphs G_{ij}. A detailed algorithm is described in [6] and [4].

4 Computational Study

We successfully applied the theoretical results to energy-efficient underground and railway timetabling, as the scheduling can be modelled as CPMC. The goal is to improve the energy efficiency of an existing schedule by choosing from different speed profiles as well as slightly shifted departure times compared to the original draft. We used real-world data provided by the operator of public transport in Nürnberg and a major German railway company respectively. In the underground application we achieved both a notable speed-up in computation times as well as an optimized schedule with 18% lower total energy consumption. This was possible with only three different speed profiles per leg and departure time shifts of $\pm 5, \pm 10, \pm 15$ s.

The implementation has been done with the Python API of the commercial state-of-the-art MIP solver Gurobi 7.0. For the computations themselves, a Six-Core AMD Opteron Processor 2435 with 2.60 GHz and 64 GB RAM has been used. Each computation has been run on six threads with a time limit of 10 h.

We optimized the schedule of line U1 of the Nürnberg underground in the morning rush-hour between 5 and 9 o'clock. Due to the integration of energy recuperation, the objective function is piecewise-linear and convex. As a result, the problem remains difficult even after adding the convex-hull description of the scheduling component. As Table 1 (left) shows, the addition of all stable-set constraints leads to a faster and better LP relaxation and outperforms the 10 h result of the intuitive formulation after just 75 min. The railway application allowed us to study the impact of using the stable-set formulation in a CPMC with an almost cycle-free dependency graph, where the impact on computation times was even more pronounced.

Figure 1 shows the total power consumption of all trains for each second (blue) as well as the accumulated energy consumption between 5 a.m. and the current time step before (red) and after optimization (green). Power consumptions below the zero line result in unused recuperated energy, which is therefore not included in the accumulated energy consumptions (red/green). The left diagram shows the power consumptions resulting from the initial schedule, whereas the right diagram shows the power consumptions of the optimized schedule.

Table 1 Nürnberg underground (left) and railway (right) instance

	Intuitive	Stable set
LP time	30.65 s	14.09 s
LP objective	15.07 MWh	15.55 MWh
Gap at 3 min	10.20%	1.63%
Gap at 75 min	3.28%	0.63%
Gap at 10 h	0.75%	0.50%

	Intuitive	Stable set
LP time	294.49 s	105.04 s
LP objective	105.43 MWh	111.23 MWh
Gap at 11 min	12.08%	<0.01%
Gap at 10 h	1.38%	–

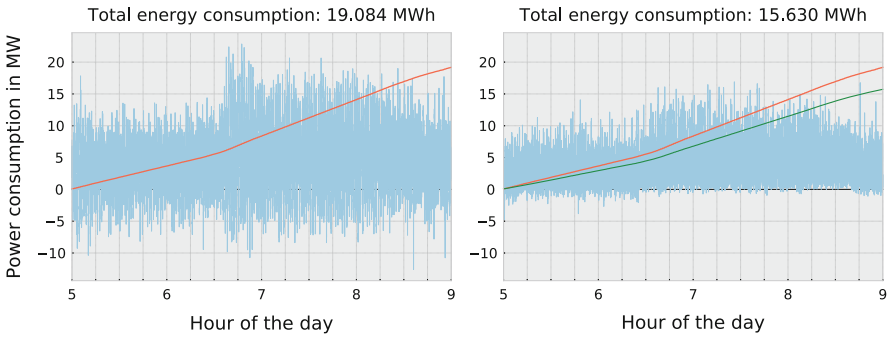


Fig. 1 Power consumption over time before (left) and after optimization (right) with total energy consumption

Table 2 Deviation from initial schedule

Total number of legs in planning horizon	2556
Legs changed to earlier departure	1144
Legs changed to later departure	1031
Legs changed to shorter travel time	99
Legs changed to longer travel time	1676
Average travel time before optimization	57.0 s
Average travel time after optimization	58.5 s

As the following table shows, this approach also keeps the impact on passenger experience at a minimum. We have a roughly even distribution of departure times being advanced and delayed with close to 15% of the departure times being unchanged. The average travel time per leg increases by about 2.6% (Table 2).

5 Conclusions

We discussed theoretical aspects of the clique problem under multiple-choice constraints with cycle-free dependency graphs (FCPMC). After showing that the clique-variant CPMC is still an $\mathcal{NP}$-hard problem, it has been determined that FCPMC is in $\mathcal{P}$ and a full description of the CPMCF polytope has been developed. Further, it has been shown that the FCPMC polytope can have exponentially many facets. We then provided criteria to more efficiently find stable set inequalities which induce facets of the CPMCF polytope, as well as an upper bound of the number thereof. Applying our theoretical results to the problem of energy-efficient timetabling for underground and railway systems, we found that exploiting the knowledge of the complete convex-hull description of FCPMC can lead to significantly lower computation times.

References

1. Avella, P., Boccia, M., Mannino, C., Vasilyev, I.: Valid inequalities for time-indexed formulations of the runway scheduling problem. In: CEUR Workshop Proceedings (2016)
2. Bärmann, A., Gellermann, T., Merkert, M., Schneider, O.: Staircase Compatibility and its Applications in Scheduling and Piecewise Linearization. Tech. Report (2016). http://www.optimization-online.org/DB_HTML/2018/01/6435.html
3. Bärmann, A., Martin, A., Schneider, O.: A comparison of performance metrics for balancing the power consumption of trains in a railway network by slight timetable adaptation. Public Transp. 1–19 (2017)
4. Bärmann, A., Gemander, P., Merkert, M.: The Clique Problem with Multiple-Choice Constraints under a Cycle-Free Dependency Graph. Tech. Report (2018). http://www.optimization-online.org/DB_HTML/2018/01/6436.html
5. Chudnovsky, M., Robertson, N., Seymour, P., Thomas, R.: The strong perfect graph theorem. Ann. Math. **164**(1), 51–229 (2006) http://www.jstor.org/stable/20159988

6. Gemander, P.: The Clique-Problem under Multiple-Choice Constraints with Cycle-Free Dependency Graphs – Structural Investigation and Application to Subway Timetabling. Master's Thesis. FAU Erlangen-Nürnberg, Erlangen (2017)
7. Sansó, B., Girard, P.: Instantaneous power peak reduction and train scheduling desynchronization in subway systems. Transplant. Sci. **31**(4), 312–323 (1997)

An Adaptive Large Neighborhood Search for Routing and Scheduling Carsharing Service Requests

Magdalena P. Lippenberger

1 Introduction

Over the past years, the importance of carsharing has been growing continuously and still develops at rapid pace. The major reason for the societies focus on mobility and new concepts therefore, is the increasing mobility need in urban areas. In this perspective, carsharing makes a great contribution to multimodal mobility concepts. The number of carsharing customers has been growing strongly over the past years. Along with this development, the amount of carsharing vehicles has been rising as well. Within 2017, the number of carsharing vehicles provided in Germany increased by 4.4% up to 17,950 vehicles (see [1]). Therefore, it is becoming more and more complex to manage a fleet as a carsharing provider. The provider needs to find an effective way to schedule service requests in order to handle increasing complexity, save costs, potentially increase revenues and reach a high service level, such that customers are satisfied. Service requests include tasks such as refueling or recharging cars, basic maintenance such as ensuring the right tire pressure, cleaning the cars, relocating the cars to locations with high demand, and many more. The requests are served by on-site agents who get to the cars either by driving another carsharing vehicle or by being dropped-off or picked-up by a colleague. Since the vehicles are spread all over the city, a good planning is required in order to reduce the time that is required to serve all requests. Thus, the providers goal is to determine efficient and applicable routes for the agents. The goal of this thesis is to develop a fleet management tool for the carsharing services DriveNow in Munich and ReachNow in Seattle, a joint venture and a subsidiary of the BMW Group, respectively. The starting point of the thesis was a situation where many steps were

M. P. Lippenberger (✉)
Technische Universität München, Munich, Germany
e-mail: magdalena.lippenberger@tum.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_2

done manually during these processes. A great improvement potential was identified in an algorithm that assists the daily life business by planning the routes covered by shuttles and agents. Therefore, the focus of this thesis is the development of a practice-oriented planning tool that supports the dispatching of the on-site agents.

2 Problem Statement and Methodology

The question that the planning tool should answer is: Which car should be serviced next, and by whom of the on-site agents? The answer depends mainly on the current location of the car and the personnel, and the urgency of the service request in combination with a time window in which the service has to be completed. Additionally, the routing of the shuttles that drop off and pick up the agents at the different locations has to be determined. After analysing the processes at DriveNow Munich and ReachNow Seattle, a framework for a specific problem scenario was defined. This problem definition includes shuttle vehicles that drop off agents at the start location of an open request. After completing the task, a shuttle picks the agents up in order to bring them to the next destination. Vehicles start and end their routes in the beginning of a shift at a depot, e.g. the headquarter, and have to return to that destination afterwards. Within the scenario, requests might differ within their assigned level of criticality. For example issues where the surrounding is affected, such as a car left at a spot where it is hindering the traffic are of higher severity and need to be processed more urgently. Therefore, requests of higher criticality level have to be preferred compared to requests of lower criticality. Based on the framework, a mathematical model was developed as a variant of a Pick-up and Delivery Problem and a Team Orienteering Problem. Thereby, we considered the special requirements like different start and ending points for the agents, the assignment to the shuttles and different levels of criticalities of requests. Considering the real-world situation of routing the agents and shuttles to handle carsharing requests, it is not trivial to define what makes a solution good or efficient, and thus optimal. Different aspects can be considered when defining a goal: the amount of requests, the level of importance of served requests, the covered travel distances and times, the compliance to time windows and fast completion of requests. For that reason, we introduce different objective functions and reflect on their properties.

3 Solution Algorithm

In order to solve the formulated problem in a low amount of time, we decided to apply a metaheuristic. We chose the Adaptive Large Neighborhood Search (ALNS) since it promises large variations by restructuring up to 30–40% of all requests within one iteration (see [3] and [2]). The Adaptive Large Neighborhood Search

(ALNS) was adapted to the problem setting and tested by applying real-world data. The analysis of different possibilities for the hierarchical objective function showed that some very practicable routes are obtained when including the minimization of the latest arrival time in the objective. Then, all shuttles are occupied evenly and the results reach a high number of served requests. Due to the fact, that drop-off and pick-up of agents do not need to be handled by the same vehicle, some adaptations had to be made to the ALNS procedure. The ALNS algorithm is initialized through a construction heuristic. Then, the goal is to improve the solution by removing and reinserting requests in each iteration. Within the removal process, a destroy operator is chosen randomly based on the operator-related weights. The destroy operator refers to simple heuristics defining the rule on which requests are removed from the current solution, e.g. a random removal or the most expensive removal heuristic which aims at removing requests that are placed in bad positions regarding travel distance. Similarly, the insertion process follows. Here, randomly a repair operator is chosen. One of these operators is the regret heuristic. The original idea is based on the assumption that the drop-off and pick-up node belonging to one request are assigned to the same route. Therefore, if a request cannot be inserted at the best combination of positions, it is assured that within the next iteration, it can definitely be inserted in the second best route. However, this idea is not entirely carried out within the adapted version of the adapted algorithm. The reason is that drop-off and pick-up node do not have to be assigned to the same route. Many combinations of adding the request evolve and can affect two different routes, if pick-up node and drop-off node are assigned to different vehicles. It cannot be assured that the second best combination of positions can be realized, if the best possible combination is not feasible anymore because another request that was added before hinders the insertion. However, the heuristic can still lead to good results. Similarly, the objective function in our problem differs compared to the ALNS algorithms presented in literature. Within the thesis, we use a hierarchical objective. It allows different choices for the objectives, such as the maximization of collected score, the minimization of the arrival times of the vehicles back at the depot or the minimization of time window exceedances. Through the hierarchical comparison of the solutions, it is determined whether the global best solution S and the currently accepted solution are updated. If the new solution S_0 is worse than the current solution, S_0 may still be accepted based on the criteria of a simulated annealing approach. At the end of each iteration, the performance of the chosen removal and insertion operators is tracked. Those statistics are necessary to update the operated-related weights at the end of each segment. This update is set in such way that higher weights result for better performing operators for the following segment. Within selection approach, higher weights refer to a higher probability of choosing the operator.

4 Computational Study

We divided the computational study into two parts. First, we set up a controlled numerical design in order to evaluate the exact solutions and to see how different objective functions influence the results. In the second part, we tested our adapted ALNS heuristic on real-world data. By solving the exact model, several insights on different behaviours of the possibilities for objectives were gained. The results showed that the secondary and tertiary objectives had a great effect on the routes of the final solution, and, therefore, on the primary objective value. The reason is that the objectives have a major impact on which requests are chosen to be added next within the insertion heuristic operation. One combination that showed useful and applicable results for the reality are the following hierarchy levels:

1. maximization of the total sum of collected score
2. minimization of the total time window exceedances
3. minimization of the arrival time of the latest vehicle at the depot T_{max}.

Within this combination, the minimization of time window exceedances encourages that requests are completed within the defined time span and Tmax assures that the resulting schedule is distributed evenly between the shuttles. The advantage here is that requests are fulfilled as fast as possible. Then, shuttles and agents might be able to fulfil additional new evolving requests later during the day. However, the choice of the objectives depends on the preferences of the fleet manager trading off between e.g. fulfilling more requests or meeting time windows. In the second part of the study, we applied our ALNS algorithm to three randomly chosen days of the real-world data sets. In order to tune the performance of the ALNS algorithm, we tested different parameter settings. We observed that there are slight differences for the advantageous settings considering the different data sets. The data sets vary mainly by the particular amount of open requests. This indicates that one factor that influences how the algorithm performs with a set of specific parameters might be the size of the data set. The amount of open requests has a major influence on the run time because more combinations of inserting and removing requests of have to be evaluated. Therefore, calculation effort increases. It stands out that most of the improvements of the solution occur within the first iterations. The proportion of new best solutions considering all found solutions within one algorithm run is rather small. Between none and fifteen new global best solutions were found within the total amounts of 400 up to 14,000 iterations. The performance was a lot better for smaller data sets of ten or 20 requests. Here, up to 15 improvements within only 10 min were found. After analysing the behaviour for different parameter settings, we evaluated different objectives and their effects on the final solution. This was done in order to see which objectives are most practicable to be applied in reality. The results show that the quality of the initial solution as well as the improved solutions depended highly on the chosen objectives for the different hierarchy levels. Based on these findings we recommend to choose the maximization of the sum of score as primary objective, the minimization of time window exceedance as

secondary objective and the minimization of the overall working time as tertiary objective.

5 Contribution and Conclusion

This masters thesis focuses on the scheduling and routing of service requests arising for vehicles of a carsharing fleet, such as relocating or refuelling cars. The formulated mathematical model achieves to depict the main constraints of the real-world scenario. Moreover, the model integrates Vehicle Routing Problem (VRP) variants that were, to the best of our knowledge, not examined in this specific combination before. To solve the model, the known ALNS algorithm was adapted in order to handle the depicted scenario with a new combination of constraints. Because of the differences to the considered problem definitions in literature, some of the remove and insertion operators needed to be adapted. We used a hierarchical objective that allows us to consider different choices for the hierarchy levels. The functionality of the developed algorithm can already be integrated within the fleet management application of the BMW Group. This will support the fleet manager and service provider to coordinate the requests efficiently, to handle complex temporal dependencies, and to prevent some manual work.

References

1. bcs. Bundesverband CarSharing. https://carsharing.de/alles-ueber-carsharing/carsharing-zahlen/aktuelle-zahlen-daten-zum-carsharing-deutschland
2. 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)
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)

Integration of Drones in Last-Mile Delivery: The Vehicle Routing Problem with Drones

Daniel Schermer

1 Introduction

The *Traveling Salesman Problem* (TSP) and the *Vehicle Routing Problem* (VRP) are well-studied problems in Operations Research (see, e.g., [12]). Most recently, in the academic literature, several researchers have shown interest in the integration of drones in last-mile delivery and several new variants of the TSP and VRP have been proposed [1, 2, 7, 13]. In particular, the idea of using drones in tandem with vehicles is commonly accepted. In this context, by allowing vehicles to be assisted by drones, we might investigate the possibility of reducing the costs or makespan. Meanwhile, in many related industries such as energy, agriculture and forestry, environmental protection, and emergency response, drones are slowly becoming a proven technology (see [8] and references therein).

In this work, we are interested in solving a new problem called the *Vehicle Routing Problem with Drones* (VRPD) [13]. The VRPD can be formulated as a *Mixed Integer Linear Program* (MILP). Furthermore, the formulation can be enhanced through the introduction of some *valid inequalities* [11]. Nevertheless, only small-scale instances can be solved optimally within a reasonable amount of time. In order to overcome this issue and address large-scale instances of the VRPD, we propose an algorithm that is based on *Variable Neighborhood Search* (VNS) and, notably, two heuristic *drone insertion* operators as local search procedures within our method. We perform an extensive computational study and conclude that the use of drones might significantly reduce the makespan. Furthermore, according to the numerical results, our algorithm is able to provide high-quality solutions in short computation time [11].

D. Schermer (✉)
Chair of Business Information Systems and Operations Research (BISOR), Technische Universität Kaiserslautern, Kaiserslautern, Germany
e-mail: daniel.schermer@wiwi.uni-kl.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_3

2 The Vehicle Routing Problem with Drones

Suppose that we have a complete graph $\mathcal{G}(V, E)$ that specifies a depot and customer locations as well as edges that connect each location to the other ones. In the VRPD, we assume that a fleet $\mathcal{K}$ of vehicles is given such that each vehicle is equipped with a set $\mathcal{D}$ of homogeneous drones. In this problem, we look for a routing of the fleet that minimizes the makespan, i.e., the time required to serve all customers by using the vehicles and the drones such that, by the end of the mission, all vehicles and drones must be at the depot [9, 13]. As drones are not subjected to the limitations of following the road network or congestion, we can assume that they are able to move faster between two locations than vehicles. However, small rotor-powered drones typically have a very limited carrying capacity. Further, as such drones rely on comparatively small batteries for powering their flight, their area of operation is much more tightly restricted compared to a vehicle that is powered by a fossil fuel (or a high-capacity battery). In the VRPD, we attempt to exploit the mutually supportive features of vehicles and drones. To this end, the drones may be launched from and retrieved by the vehicle at any vertex. When launched, a drone can travel a limited distance and serve exactly one customer. After making the delivery, the drone must return to the same vehicle from which it was launched to be resupplied. In the VRPD, the time required to launch or retrieve a drone and the time that is needed to serve a customer are assumed to be negligible. Furthermore, we require that the drone's battery is recharged (or swapped) instantaneously once it has returned to the vehicle. Through this tightly coupled interaction between vehicles and drones, the VRPD is concerned with several aspects of *synchronization* that are highly relevant in vehicle routing [3].

3 Solution Methods

Using the MILP formulation of the VRPD, we can use any standard MILP solver, e.g., IBM CPLEX or Gurobi Optimizer, to tackle the model. However, due to the complexity of this problem, only small-sized instances can be solved to optimality within reasonable time. Through the introduction of valid inequalities, we can improve the performance of the solvers [11]; however, this is not sufficient for solving large VRPD instances.

Thus, in order to address large-scale VRPD instances, we introduce an algorithm that is based on *Variable Neighborhood Search* (VNS), a metaheuristic that was proposed by Mladenović and Hansen [6] for solving complex optimization problems. VNS provides a framework for a systematic change of neighborhoods within a *Local Search* (LS) approach. In this framework, VNS favors characteristics of the incumbent solution while exploring increasingly distant neighborhoods in its attempt to escape local optima. Among several variants of VNS, *Variable Neighborhood Decomposition Search* (VNDS), introduced by Hansen et al. [4],

extends the basic VNS into a nested two-level VNS scheme that is built on a decomposition of the problem into traceable subproblems.

Based on our preliminary results [10], we selected VNDS for investigating its performance in solving large-scale VRPD instances. Through this approach, the problem can be decomposed into the determination of the vehicle's routes and the insertion of drone operations. For this purpose, after a deep analysis of the VRPD, we introduce some operators for generating new neighborhood structures that permit us to explore the solution space of a given VRPD instance. Furthermore, we develop two new problem specific local search operators, called *greedy insertion heuristic* (GIH) and *savings insertion heuristic* (SIH), that are used to place drones into existing routes of vehicles [11]. As the objective value in a VRPD solution is determined by the longest path through a directed acyclic graph, both operators attempt to reduce the length of this path by inserting feasible drone operations. Whereas the GIH continuously attempts to reduce the length by inserting the *first feasible* improving move, the SIH relies on a heuristic estimator to continuously find the *best possible* improving move until all drones have been placed and no more feasible (or improving) moves exist. Let y and x be feasible solutions to a VRP* (to which VRPD constraints apply, except that no drones can be used) and VRPD, respectively. Given an initial solution y (i.e., a routing of vehicles where no drone operations occur), VNDS (Algorithm 1) works as follows:

- First, we find a VRP* solution y that favors characteristics of the current incumbent VRPD solution x. In the first iteration, we take the solution y that was found through an initialization procedure. For the remaining iterations, we take the solution $y(x)$ that was stored after a neighborhood change.
- Then, we find a solution $y' \in \mathcal{N}_k(y)$ (where $\mathcal{N}_k(y)$ is the k_{th} neighborhood of y) to a VRP* through basic VNS:
  - During VNS, first we apply a *shake* to generate a *random* solution y' that is located in $\mathcal{N}_k(y)$ [6].
  - Then, we apply a local search procedure to improve y' using *2-opt* [5], *String Relocation* and *String Exchange* [12].
  - The inner VNS, that generates y', continues until reaching a predefined limit of maximum iterations i_{max}.

Algorithm 1 VNDS($\mathcal{G}, \mathcal{K}, \mathcal{D}, k_{max}, i_{max}, t_{max}$)

```
repeat
    k ← 1;
    repeat
        y ← Shake(y(x), k);
        y' ← VNS(y, k, K, i_max);
        x' ← DroneInsertion(y', D);
        x, y(x), k ← NeighborhoodChange(x, x'y', k);
    until k = k_max
until t > t_max
return x;
```

- Once we have found a VRP* solution y', we transform it to a VRPD solution x by inserting drones into the existing route of each vehicle.
- Afterwards, the neighborhood change operator updates the values of x, $y(x)$ (in case of a new incumbent solution) and k. The algorithm continues until reaching a run time limit t_{max}.

4 Computational Study

We perform an extensive computational study to assess the performance of our proposed heuristic with different combinations of vehicles and drones per vehicle. Further, we differentiate between three different drone endurance parameters (that determine the battery capacity) and relative velocities $\alpha = \overline{v}/v \in \{2, 3, 4\}$, where $\overline{v}$ and v are the velocities of the drones and vehicles, respectively. We use small-scale (11 vertices) and large-scale instances (up to 532 vertices) and solve them through Gurobi, using the MILP formulation, and VNDS, respectively [11].

Overall, our first observation from the numerical results states that the use of drones can have a significantly positive impact on the reduction of the makespan. Figure 1 shows how the number of vehicles, drones per vehicle, and relative velocity affect the makespan on a large-scale instance solved through VNDS.

Here, we can also highlight the decreasing marginal benefit of each additional drone per vehicle (see, e.g., $|\mathcal{K}| = 1, |\mathcal{D}| = 1$ and $|\mathcal{K}| = 1, |\mathcal{D}| = 2$). In the case of multiple drones per vehicle, they must be increasingly fast or have a higher endurance (i.e., battery capacity) to be utilized effectively in most cases. However, after reaching a certain threshold, no further improvement is possible

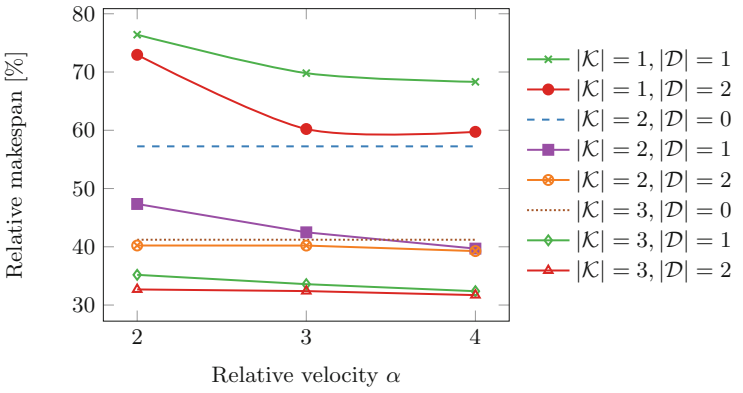


Fig. 1 Relative makespan in VRPD solutions on an instance with 150 customers depending on the number of vehicles $|\mathcal{K}|$, drones per vehicle $|\mathcal{D}|$, and relative velocity α for a fixed drone endurance. The makespans are shown relative to that of a single vehicle equipped with no drone ($|\mathcal{K}| = 1, |\mathcal{D}| = 0$)

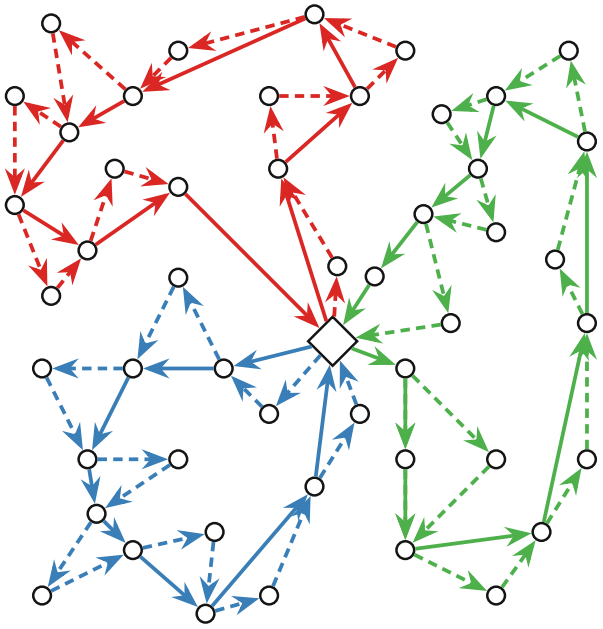


Fig. 2 A sample solution for an instance with 50 customers, three vehicles, and one drone per vehicle. The vehicles' and drones' paths are shown using solid and dashed lines, respectively

by increasing the relative velocity or endurance of the drones. In these cases, the longest path is likely to be defined by the route of the vehicle. We can show that high-quality solutions can be achieved in short computation time. Compared to the results that were found during our preliminary study (see [10]), our algorithm performs substantially better [11]. Figure 2 shows a sample VRPD solution.

5 Concluding Remarks and Perspectives

This study addresses the VRPD and investigates its complexity through computational experiments. According to the numerical results, small instances might be solved by any standard MILP solver. In order to address large-scale instances within reasonable time, we proposed a heuristic method. For this purpose, we introduced a VNS-based algorithm that is able to solve large-scale VRPD instances in short computation time while providing high-quality solutions.

As an emerging topic, the integration of drones in last-mile delivery provides interesting optimization problems that are also challenging from computational point of view. Hence, in this context, there is a vast research avenue that needs to be explored. In particular, it will be of interest to have a good estimation of the potential benefits in terms of makespan or cost before drones might be included in

practical applications. In our work, we provide a clear indication that drones might be beneficial with regards to a reduced makespan. However, no particular focus was given to cost effects. Additionally, the potentially different quality of service was neglected and drones were assumed to operate in a risk-free environment (i.e., a delivery will always be possible and succeed). Taking these factors into account needs investigation of new optimization problems. In particular, future research might be dedicated, but not limited, to addressing more realistic models, e.g., accepting the possibility of launching and recovering drones from arbitrary grid points or introducing more accurate battery models. We have the objective of exploring these more realistic and complex problems and the results will be published in the future.

Acknowledgements I am grateful to the advisers of my Master Thesis, i.e., Mahdi Moeini and Oliver Wendt, for introducing me to this exciting topic and for their continued support.

References

1. Agatz, N., Bouman, P., Schmidt, M.: Optimization approaches for the traveling salesman problem with drone. Transp. Sci. **52**(4), 951–981 (2018)
2. Di Puglia Pugliese, L., Guerriero, F.: Last-mile deliveries by using drones and classical vehicles. Optim. Decis. Sci. **217**, 556–565 (2017)
3. Drexl, M.: Synchronization in vehicle routing—a survey of VRPs with multiple synchronization constraints.Transp. Sci. **46**(3), 297–316 (2012)
4. Hansen, P., Mladenović, N., Perez-Britos, D.: Variable neighborhood decomposition search. J. Heuristics **7**(4), 335–350 (2001)
5. Lin, S., Kernighan, B.W.: An effective Heuristic Algorithm for the traveling-salesman problem. Oper. Res. **21**(2), 498–516 (1973)
6. Mladenović, N., Hansen, P.: Variable neighborhood search. Comput. Oper. Res. **24**(11), 1097–1100 (1997)
7. Murray, C.C., Chu, A.G.: The flying sidekick traveling salesman problem: optimization of drone-assisted parcel delivery. Transp. Res. C Emerg. Technol. **54**, 86–109 (2015)
8. Otto, A., Agatz, N., Campbell, J., Golden, B., Pesch, E.: Optimization approaches for civil applications of unmanned aerial vehicles (UAVs) or aerial drones: a survey. Networks **72**(4), 411–458 (2018)
9. Poikonen, S., Wang, X., Golden, B.: The vehicle routing problem with drones: extended models and connections. Networks **70**(1), 34–43 (2017)
10. Schermer, D., Moeini, M., Wendt, O.: Algorithms for solving the Vehicle Routing Problem with drones. Lecture Notes in Artificial Intelligence (Springer) **10751**, 352–361 (2018)
11. Schermer, D., Moeini, M., Wendt, O.: A Variable Neighborhood Search Algorithm for Solving the Vehicle Routing Problem with Drones, Technical Report, p. 33. Chair of Business Information Systems & Operations Research, Technische Universität Kaiserslautern, Kaiserslautern (2018)
12. Toth, P., Vigo, D.: The Vehicle Routing Problem. Society for Industrial and Applied Mathematics, Philadelphia (2002)
13. Wang, X., Poikonen, S., Golden, B.: The vehicle routing problem with drones: several worst-case results. Optim. Lett. **11**(4), 679–697 (2016)

Robust Evacuation Planning for Urban Areas

Marc Maiwald

1 Motivation

Disaster management has become an important topic due to the growing frequency and intensity of natural and man-made disasters. The reasons for this negative trend are various: Environmental problems such as global warming cause bush fires or floods. Poor planning of buildings or cities increases the consequences of hurricanes or earthquakes. Moreover, humans themselves are intentionally (e.g., terrorist attacks) or unintentionally (e.g., chemical disasters) responsible for a multitude of disasters. The Federal Emergency Management Agency of the United States reports annually 45–75 disasters in recent years that require state and federal assistance and may lead to an evacuation [2]. Furthermore, the United Nation Office for Disaster Risk Reduction reported for the years 1995–2015 that over 606,000 people were killed and over four billion people were affected by natural disasters worldwide [6].

This work focuses on the mass evacuation of urban areas. The two major issues faced during such an evacuation scenario are the following: First, evacuees lack both information and experience regarding such an extreme situation. Second, people might act in a selfish way and choose a route which is the best for themselves. Thus, they might choose an evacuation route which might be the best route during normal traffic, but not during an extraordinary evacuation situation. The main task of evacuation planning is the guidance of the evacuees through the street network to reduce casualty risks and increase the performance of the evacuation process. The guidance includes instructions to evacuees when to leave their homes and start their evacuation process, which evacuation routes to take and which shelter to head to.

M. Maiwald (✉)
University Duisburg-Essen, Duisburg, Germany
e-mail: Marc.Maiwald@uni-due.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_4

Careful and rigorous evacuation planning is an important task to ensure an effective implementation of the evacuation process and to save the life and health of the exposed population.

One major issue of this research field is the estimation of realistic scenarios as the relevant input data is highly stochastic. We focus in this work on the consideration of uncertainties regarding the supply conditions of the street network during an evacuation process. These supply variations might be caused by the catastrophe itself (e.g., (partial) flooding of streets, limited visibility or debris on the road) or by factors that also affect daily traffic such as a wide variety of traffic accidents or technical failures. The current evacuation literature usually assumes the identical street capacities for an evacuation scenario as for daily traffic situations which is in fact a quite unrealistic assumption. One consequence of these reduced capacities is a lower traffic volume that can be transferred by the streets. This is even worse for urban evacuation scenarios as the main evacuation routes are mostly utilized up to their capacity limit. Mattson and Jenelius [4] state that traffic systems become less robust (= more vulnerable) if the capacity utilization is increased. Therefore, it is important to anticipate these supply condition uncertainties in the urban evacuation planning process.

2 Balanced Street Network Utilization for Increasing Robustness

The contribution of this work lies in designing a deterministic optimization model that is more robust against supply uncertainties in the street network without the explicit definition of these uncertainties. On the modelling level no explicit index exists that affects or improves the robustness of a traffic assignment. Therefore, we try to adopt an idea that has already been successfully applied among others to robust network design: The robustness of street networks is enhanced by a better utilization of the available network capacities and by reducing interdependencies in the network [5]. The general idea is to utilize the available street capacities in a more balanced way to reduce the consequences of disruptions in the street network during the evacuation process. Therefore, we extend the initial evacuation model basing on the Cell-Transmission model [1] by considering the utilization of the street network during the evacuation process. The resulting model has two new properties compared to the initial evacuation model: First of all, it bases on predefined evacuation paths. The consideration of these evacuation paths is necessary to prevent unreasonable routes induced by utilizing the network capacities in a more balanced way. Unreasonable routes include e.g., circles or utilize streets in both directions, which is normally not meaningful in an evacuation scenario. Therefore, we also present a special path generation algorithm to generate a reasonable number of paths meeting our individual defined requirements. Secondly, the new model includes two objective functions to handle the evacuation performance as well as

the robust aspect. For handling these two objectives we apply the lexicographic ϵ-Constraint Method. The main following research objective is to answer the question if the robustness of the evacuation guidance can be increased if the street network capacities are utilized in a more balanced way. Evacuation performance (e.g., the total evacuation time) is not our only objective and we are willing to sacrifice some of it to enhance the robustness in the face of unpredictable capacity disruptions.

3 Approximated Pareto Sets

The lexicographic ϵ-Constraint Method gives us no information about the trade-off concerning the evacuation performance and the robust aspect. Therefore, we present two approaches approximating the optimal Pareto set. The advantage of approximated Pareto sets is that not the complete optimal Pareto set has to be determined which is usually computational burdensome. The first approach is the box algorithm which is based on the work of [3]. In this approach a coordinate system is considered where each axes represent one objective function. One solution of the extended evacuation model is specified by the two objective function values so that each solution is represented by one definite point in the coordinate system. We determine for each adjacent solution point pair a rectangle that represents the solution space of all non-dominated solutions between these points. These rectangles are iteratively divided into smaller rectangles by this approach. The objective of this approach is to reduce the total area of all rectangles. The smaller the total area of all rectangles, the better the approximation of the optimal Pareto set. This approach is based on optimal solutions so that it may suffer from long computational times. Thus, we additionally present the staircase algorithm. The general idea of this second approach is identical to the box algorithm: We consider the entire solution space and reduce this space by generating more solutions iteratively. In contrast to the box algorithm we consider lower and upper bounds instead of optimal Pareto solutions. Thus, the solution quality of one iteration of the staircase algorithm is usually lower than in one iteration of the box algorithm, but the computational times of one iteration can be reduced enormously. Both bounds are determined by individual problem-specific heuristics. Figure 1 shows two typical results for each algorithm to visualize the general differences of the approaches. The points p_1–p_5 represent each one optimal solution and the points lb_1–lb_5, respectively, ub_1–ub_5 represent each one solution of the individual bounds. Thus, the grey marked area visualizes the solution space of the approximated Pareto set. The small the area, the better is the approximated Pareto set.

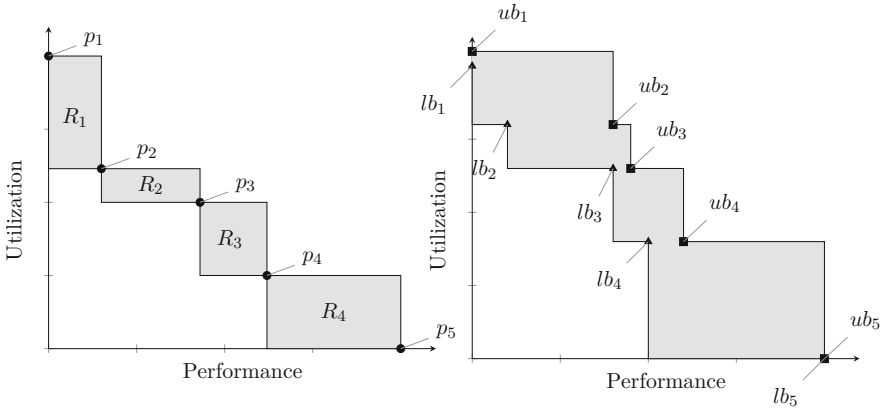


Fig. 1 Typical solutions of the box-algorithm (left side) and the staircase algorithm (right side)

4 Computational Results

We conduct an extensive computational study that is subdivided into three single studies. First, we investigate the effects of various parameter settings regarding the evacuation performance and the robust aspect. Based on these results, we arrange a test bed for the upcoming two studies. In the second study we compare the performance of both approaches approximating Pareto sets. Thirdly, we test the benefits of the robust evacuation concept with the help of our individual test procedure.

By evaluating the various parameter settings we detect that the population size, the distribution of the evacuees within the networks and the tolerance factor, which determines the predefined set of evacuation paths, mainly affect the evacuation performance and the robust aspect. Thus, we design a test bed consisting of 252 instances mainly varying in these parameters. The main results of the computational studies are the following: By comparing both solution approaches the staircase algorithm archives in $80.\bar{5}\%$ of all considered instances a better solution quality than the box algorithm. In the average the staircase algorithm represents the solution space 31.8% better than the box algorithm. This result confirms the general idea of the staircase algorithm as meaningful. By evaluating the benefits of the robust concept, we observe that a more balanced capacity utilization does not naturally lead to a more robust solution. However, we detect that the lexicographic solution, considering the optimal evacuation performance and the corresponding best balanced network utilization, performs always better or at least as well as if only the evacuation performance is considered. This result confirms our concept as beneficial and we recommend to apply at least this lexicographic solution, because no additional hazard has to be accepted. However, the absolute benefits of these solutions are marginally, but they can be increased if an additional hazard is accepted. Unfortunately, we cannot give a general advise for the exact level of

additional hazard. The reason is that a more balanced capacity utilization naturally leads to a higher accepted hazard level compared to the optimum which is harder to catch up in the average. Actually, there is a (small) range in the additional accepted hazard level in which the robust concept performs better than the safe concept. Unfortunately, this range depends on the unknown scenario and the unknown capacity disruption(-s) (levels). Nevertheless, the computational study shows that several parameter settings affect the performance of the robust concept positively. Thus, the benefits of the robust concept naturally increase if higher capacity disruption levels or multiple capacity disruptions occur at once. We also detect that the more unbalanced the vehicles are initially distributed within the networks, the more beneficial is the robust concept. This is due to the fact that the potential to utilize the network capacities in a more balanced way is the greatest for these scenarios. Interestingly, the results of the staircase algorithm are also the best for these instances. Nevertheless, we cannot clearly predict the benefits of the robust concept for a given evacuation scenario, but we can at least partially estimate whether the robust concept is worthwhile or not.

5 Conclusion and Future Research

We focus on short-term uncertainties in the street network capacities and implement the robust aspect in a unique way: We consider a deterministic cell-based evacuation model that sacrifices a certain amount of the optimal evacuation performance to utilize the street capacities in a more balanced way. Thus, we reduce the consequences of disruptions and design the evacuation plan more robust. The results are evaluated in a computational study and show that the resulting evacuation plans are (in the average) less sensitive to disruptions than the evacuation plans only focusing on the performance.

Future research can be conducted in many fields: First of all, we investigate various characteristics regarding 16 networks that differ, e.g., in the network size or network structure. However, we cannot determine any significant variations in the computational study results among these networks, although we suppose that some network characteristics affect the evacuation planning concept. Thus, the investigation of these network characteristics is an interesting starting point for future research projects. Additionally, no universal test bed exists for cell-based evacuation planning approaches. Thus, the arrangement of a comprehensive test bed that comprises a multitude of the mentioned characteristics will help to compare various cell-based evacuation approaches in the future. Here, the results of our test bed generation might help.

Second, our results of problem-specific lower and upper bounds are inconstant. In the average the gap between the lower and upper bound is about 6.7% in our computational study, but we detect variations in the results regarding various parameter settings. Thus, this result is also a starting point to improve or extend

the computation of the bounds. Perhaps, also the investigation of other solution approaches is purposeful.

References

1. Daganzo, C.F.: The cell transmission model: a simple dynamic representation of highway traffic. Transport. Res. B-MEH **28**, 269–287 (1994)
2. Guha-Sapir, D., Vos, F., Below, R., Penserre, S.: Annual Disaster Statistical Review 2011: The Numbers and Trends. UCL, Bloomsbury (2012)
3. Hamacher, H.-W., Pedersen, C., Ruzika, S.: Multiple objective minimum cost flow problems: a review. Eur. J. Oper. Res. **176**, 1404–1422 (2007)
4. Mattsson, L.-G., Jenelis, E.: Vulnerability and resilience of transport systems: a discussion of recent research. Transport. Res. A-POL **81**, 16–34 (2015)
5. Snelder, M., Immers, B., Van Zuylen, H.: The best of two worlds: a robust road network design method based on an optimization model and expert judgement. In: 5th International Symposium on Transportation Network Reliability, INSTR, Hong Kong (2012)
6. United Nations Office for Disaster Risk Reduction (UNISDR) and Centre for Research on the Epidemiology of Disasters (CRED): The Human Cost of Weather–Related Disasters, 1995–2015. United Nations, Geneva (2015)

Contributions to Branch-and-Price-and-Cut Algorithms for Routing Problems

Ann-Kathrin Rothenbächer

1 Introduction

Branch-and-price-and-cut algorithms are powerful approaches for solving combinatorial optimization problems such as routing problems to optimality [1]. These algorithms comprise the techniques branch-and-bound, column generation, and addition of valid inequalities, which are interdependent. The branch-and-bound method ensures feasible integer solutions through a systematic and usefully restricted enumeration. Column generation is applied to formulations with a huge number of variables to find and add improving columns dynamically. In routing problems, the problem of finding improving columns is usually an elementary shortest path problem with resource constraints (ESPPRC) [4]. Valid inequalities (cuts) can be identified and added to a formulation to strengthen its linear relaxation. The details of all three parts depend on the problem that is tackled.

Routing refers to the task of selecting paths through a network. A very famous routing problem is the vehicle routing problem (VRP) where a fleet of vehicles has to perform a set of tasks with minimal costs [10]. There are numerous variants of the vehicle routing problem. In the cumulative dissertation of the author, which is summarized in the following, applications of and improvements on branch-and-price-and-cut algorithms for routing problems will be discussed.

A.-K. Rothenbächer (✉)
Chair of Logistics Management, Gutenberg School of Management and Economics, Johannes Gutenberg University, Mainz, Germany
e-mail: ann-kathrin.rothenbaecher@deutschebahn.com; http://logistik.bwl.uni-mainz.de/

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_5

2 Dynamic Half-Way Points in Bidirectional Labeling

As mentioned above, the subproblem of routing problems within column generation is usually an ESPPRC. Here, the task is to find the best path (with respect to the reduced costs) from a given source to a given sink that fulfills a set of constraints defined over a set of resources. Although the ESPPRC is known to be NP-hard, it can be solved effectively with a dynamic-programming based labeling algorithm. The labeling can usually be performed through a forward or a backward labeling approach, but monodirectional labelings typically suffer from an explosion of labels.

Bounded bidirectional labeling was introduced in order to mitigate the explosion of labels [5]. Both forward partial paths and backward partial paths are created, but processed only up to a so-called *half-way point* of a monotone resource in the middle of its feasible domain. When labeling terminates, suitable forward and backward labels have to be merged to complete paths.

However, within asymmetric instances or in applications in which forward and backward labeling differs significantly, a predefined half-way point in the middle may be unfavorable.

In Fig. 1, an example of an ESPPRC instance with unbalanced forward and backward labeling is depicted. The functions show, for any value T of the monotone resource, the number of processed forward labels with value $\leq T$, the number of processed backward labels with value $\geq T$, and the sum of both values.

Obviously, the static half-way point H^{St} produces much more labels than necessary. In our paper [9], we present a strategy to determine the half-way point H^{Dy} dynamically by always continuing the labeling in the direction with the smaller number of unprocessed labels. As in the example in the figure, this dynamic half-way point leads to a reduced number of processed labels, which normally corresponds to a faster computation time. Much more computational results on different types of vehicle routing problems can be found in our paper [9].

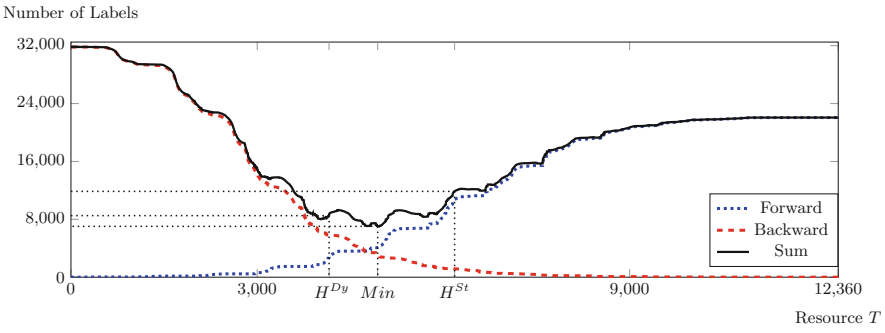


Fig. 1 Example of an SPPRC instance with unbalanced forward and backward labeling; static half-way point $H^{St} = 6180$ in the middle of the time horizon and dynamic half-way point $H^{Dy} = 4158$

3 Bidirectional Labeling for Pickup-and-Delivery

The pickup-and-delivery problem (PDP) deals with requests of transporting goods or passengers from specific pickup locations to corresponding delivery locations. Thus, compared to the classic vehicle routing problem (VRP), there are additional pairing and precedence constraints on the locations that need to be visited. We present the first bidirectional labeling approach for column-generation subproblems of PDPs in [3]. A strong dominance relation can only be achieved when it is never beneficial in terms of reduced costs to visit the second location of a request (the delivery point in forward direction and the pickup point in backward direction). However, the dual costs can not be distributed on the arcs such that this condition is true in both directions. For that reason, bidirectional search seemed to be unattractive for ESPPRCs with a pickup and delivery structure. We show that it is possible to use different cost matrices for the two directions when the merge procedure is adapted accordingly. Thus, the strong dominance relation can be used in both directions making the bidirectional search attractive again.

The performance profiles in Fig. 2 compare six labeling strategies on 220 PDPTW benchmark instances: *Fw* for forward, *Bw* for backward, *Bi* for bidirectional, *Dy* for dynamic half-way points, *St* for static half-way points, *S* for strong dominance and *W* for weak dominance. The performance profile $\rho_A(\tau)$ of an algorithm A of a set of algorithms $\mathcal{A}$ is the fraction of instances that algorithm A can solve within a factor τ of the fastest algorithm of $\mathcal{A}$. The figure shows that the branch-and-price-and-cut with labeling strategy *Bi-Dy-SS* clearly dominates all

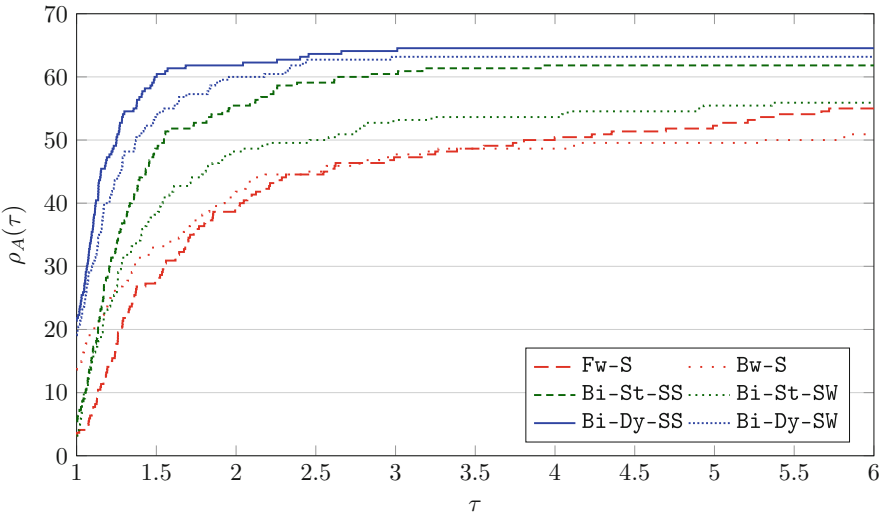


Fig. 2 Performance profiles of branch-and-price-and-cut algorithms using six different labeling strategies

other algorithms. Indeed, it is at most 1.5 times slower than the fastest algorithm for more than 90% of the instances solved by at least one algorithm.

4 Truck-and-Trailer Routing Problem

The truck-and-trailer routing problem (TTRP) describes situations where trucks and attached trailers are used to visit customers with the restriction that some customers are not reachable by trailers. Hence, trailers can be parked at some locations and have to be recoupled later. Our algorithm described in [8] combines several state-of-the-art techniques for VRPs and tackles two new extensions. The first problem extension is the enlargement of the planning horizon from 1 to 2 days. Thereby, a symmetry arises, which is tackled by a tailored column-generation stabilization method. The second problem extension is the consideration of quantity-dependent transfer times for the load transfers from the truck to the trailer. We propose an adapted labeling algorithm that can handle the emerging tradeoff between the consumed time and the free space in the truck through additional resources. The resulting trade-off curves along a simple example route of a truck-and-trailer combination are shown in Fig. 3. The vehicle starts at the depot, parks the trailer at $d(p)$, visits a lorry customer 1, drives back to the parking station for transfer $\tau(p)$, visits another lorry customer 2, picks up the trailer at $c(p)$ and finally returns to the depot.

The computational results demonstrate that our algorithm outperforms existing approaches on TTRP benchmark instances. We attribute this success mainly to the use of subset-row cuts and the new bidirectional labeling. The results regarding the extensions demonstrated their influences on the overall routing costs and showed that the solvability does not change excessively despite the more complicated problems.

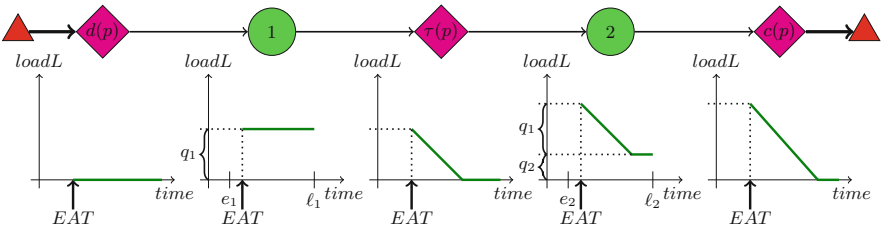


Fig. 3 Example for tradeoff propagation with quantity-dependent transfer times. EAT = earliest arrival time, loadL = load in the lorry, q_i supply and $[e_i, l_i]$ time window of customer i

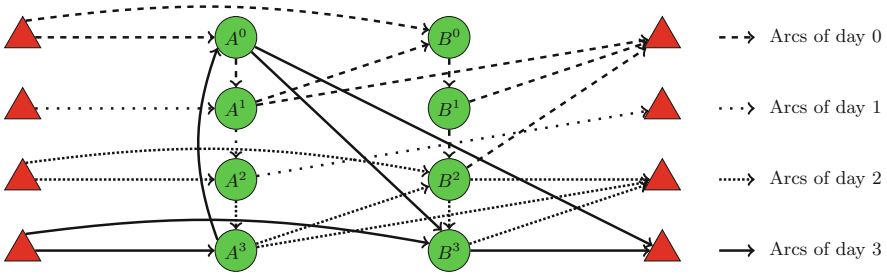


Fig. 4 Task networks for 4 days of an example with customer A offering schedules {0, 2} & {1, 3}, and customer B offering schedules {0, 2} & {0, 3} & {0, 2, 3}

5 Periodic Vehicle Routing Problem

In the periodic VRP (PVRP), the planning horizon is extended to multiple periods and customers need to be visited several times according to one of their offered visiting patterns. In the standard variant, the customers need to be visited regularly with a preset visit frequency. An exact method dealing with some more flexible visiting patterns is presented in [2].

In [6], we present a branch-and-price-and-cut algorithm for the classical PVRP with time windows and an extension to even more different visiting patterns. Whereas the classical PVRP expects visiting patterns with regular intervals between the visits and a fixed visit frequency for every customer during the planning horizon, this paper enables all kinds of visiting patterns. Therefor, we introduce two new types of underlying pricing networks, on which a state-of-the-art labeling algorithm can find new columns. The more effective network type is illustrated for a small example with two customers offering different schedules in a planning horizon of 4 days in Fig. 4.

Moreover, for instances that fulfill a special symmetry, we show that constraint aggregation significantly improves the solution process. Different PVRP-specific settings are evaluated by computational tests and the optimality of two PVRPTW benchmark instances can be proven for the first time.

6 Service Network Design and Hub Location Problem

The service network design and hub location problem (SNDHLP) tackled in [7] does not belong to VRPs, but it also includes routing decisions. Its task is to select locations and connections between them to establish a network for the routing of a given set of transport requests.

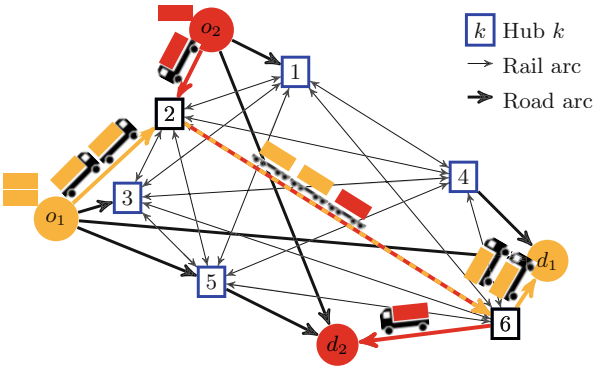


Fig. 5 Example SNDHLP instance with solution

Requests have to be transported via road and rail through a network whose hub nodes and connections between them have to be selected. In contrast to most other related works, the decision on a connection is not only about its existence but about the corresponding frequency of services. A simple example with an optimal solution is depicted in Fig. 5.

Furthermore, several other real-world conditions such as restricted number of transshipments, transport time limits and splittable requests are considered in this combination for the first time. The branch-and-price-and-cut algorithm is adapted to the problem by a specific layered pricing network, an hierarchical branching scheme, the incorporation of three types of valid inequalities, and further acceleration techniques. Computational tests on real world instances of the major German rail company prove the practical applicability of the algorithm.

7 Conclusions

All summarized papers deal with branch-and-price-and-cut algorithms to find optimal solutions for different routing problems. Thereby, existing procedures were improved and new problem variants became solvable. The bidirectional labeling for the solution of ESPPRCs was accelerated through the introduced dynamic halfway point. By using different cost matrices in forward and backward labeling, bidirectional labeling became applicable for all pickup and delivery problems. Truck-and-trailer routing problems were extended to a 2 days planning horizon and quantity-dependent transfer times. For periodic vehicle routing problems, an exact algorithm which can tackle all kinds of visiting patterns was presented for the first time. Finally, a real-world problem combining service network design and hub location in the context of combined transport was solved.

References

1. Desrosiers, J., Lübbecke, M.: Branch-price-and-cut algorithms. In: Wiley Encyclopedia of Operations Research and Management Science. John Wiley & Sons, Inc., Hoboken (2011)
2. Francis, P., Smilowitz, K., Tzur, M.: The period vehicle routing problem with service choice. Transp. Sci. **40**(4), 439–454 (2006)
3. Gschwind, T., Irnich, S., Rothenbächer, A.K., Tilk, C.: Bidirectional labeling in column-generation algorithms for pickup-and-delivery problems. Eur. J. Oper. Res. **266**(2), 521–530 (2018)
4. Irnich, S., Desaulniers, G.: Shortest path problems with resource constraints. In: Desaulniers, G., Desrosiers, J., Solomon, M. (eds.) Column Generation, pp. 33–65. Springer, New York (2005)
5. Righini, G., Salani, M.: Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. Discret. Optim. **3**(3), 255–273 (2006)
6. Rothenbächer, A.K.: Branch-and-price-and-cut for the periodic vehicle routing problem with flexible schedule structures. Transp. Sci. https://doi.org/10.1287/trsc.2018.0855
7. Rothenbächer, A.K., Drexl, M., Irnich, S.: Branch-and-price-and-cut for a service network design and hub location problem. Eur. J. Oper. Res. **255**(3), 935–947 (2016)
8. Rothenbächer, A.K., Drexl, M., Irnich, S.: Branch-and-price-and-cut for the truck-and-trailer routing problem with time windows. Transp. Sci. **52**(5), 1174–1190 (2018)
9. Tilk, C., Rothenbächer, A.K., Gschwind, T., Irnich, S.: Asymmetry matters: dynamic half-way points in bidirectional labeling for solving shortest path problems with resource constraints faster. Eur. J. Oper. Res. **261**(2), 530–539 (2017)
10. Toth, P., Vigo, D. (eds.): Vehicle Routing. Society for Industrial and Applied Mathematics, Philadelphia (2014)

Logistics Networks with Intermediate Stops: Designing Innovative and Green Solutions

Maximilian Schiffer

Current challenges and future requirements for logistics networks like tremendous growth in small package shipping and extremely ambitious ecological targets at global and local level call for more flexibility, efficiency, and sustainability in logistics. Consequently, new concepts, network structures, and technologies must be integrated in today's logistics networks. Herein, logistics networks with intermediate stops (LNWIS) constitute a promising concept to realize these needs. In LNWIS, vehicles perform additional stops on routes that are not related to customer visits. Instead, these stops are mandatory to keep the vehicle operational and are often only possible at so called intra-route facilities. These facilities are neither depots nor hubs, and are planned on the same echelon as the customer, although in some cases a second echelon might be necessary to replenish these facilities.

LNWIS offer multiple options to realize sustainable logistics networks for several reasons: first, they allow for freight replenishment in between service stops such that smaller, more flexible and more environmentally friendly vehicles can be used in city logistics. Second, intermediate stops can also be used to realize multi-modal transportation. Third, intermediate stops enable recharging of electric vehicles on routes and thus help to foster the deployment of environmentally friendly electric commercial vehicles (ECVs) in mid-haul transportation.

Against this background, this dissertation focuses on the strategic design and operation of logistics networks with intermediate stops. Herein, a generic algorithmic as well as a modeling and solution framework is derived for the design and operation of these networks. In this course, this dissertation introduces a new problem class, the LRPIF. The derived algorithmic framework can be applied to any kind of network with intermediate stops, and thus the improvement potential

M. Schiffer (✉)
TUM School of Management, Technical University of Munich, Munich, Germany
e-mail: schiffer@tum.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_6

of interdependent network location and vehicle routing decisions can be levied for any application case. Besides providing a new state of the art in modeling and solution methods for both VRPISs and LRPIFs, this dissertation shows the benefit of such concepts by applying the proposed methodology to real-world problems. As result, this dissertation succeeds in deriving managerial insights that can be used by practitioners as well as by researchers for further work.

Besides an introduction and a conclusion, this thesis comprises an extensive literature review on VRPISs and LRPIFs and two parts. The first part contains four publications on application cases in the field of electrified mid-haul logistics, while the second part comprises three methodological publications on a generic algorithmic framework for VRPISs and LRPIFs. The remainder of this article summarizes its content in the following.

1 Recent Research

LNWIS have been addressed from researchers in several application fields, spanning from snow plowing and road painting over city logistics and electric vehicles. In general, these works can be separated into three different streams: intermediate stops for (1) replenishment (e.g., freight replenishment), (2) refueling (e.g., battery recharging), and (3) *idling* (e.g., driver breaks). In the last years, there has been an increasing interest in these topics, especially regarding intermediate stops in the course of refueling, i.e., recharging electric vehicles [1]. However, this research has mostly been focusing on intermediate stops in vehicle routing problems (VRPs) from an operational perspective.

As soon as intra-route facilities are necessary to enable intermediate stops, the strategic intra-route facility location and the operational vehicle routing decision become interdependent. This interdependency has not been studied so far, as it only allows for more efficient networks in the course of replenishment stops but is not mandatory for the general network operation. With researchers and practitioners paying increasing attention to electric vehicles in logistics networks, this interdependent decision remains no longer optional but becomes mandatory as a sophisticated network design for charging stations is necessary to guarantee a faultless operation of the network.

2 Electric Commercial Vehicles in Mid-Haul Logistics

The first part of this thesis contains four articles that focus on a specific application case, electrified mid-haul logistics networks. This part of the thesis shows the benefit of the optimized planning of logistics networks with intermediate stops and derives managerial insights for fleet operators. Consequently, it integrates

practical requirements into applicable planning models that are then applied to a real-world case study.

At strategic level, deterministic as well as robust LRPs are developed in order to design optimal logistics networks, herein considering interdependencies between electric vehicle routing and charging station location decisions. At operational level, the synchronization of idle times resulting from charging processes and driver breaks due to hours of service (HOS) regulations is addressed to minimize time inefficiencies of ECVs in mid-haul transportation. Both the strategic and operational planning approaches succeed in minimizing the limitations of ECVs within logistics fleets considerably, thus paving the way for the market diffusion of electric vehicles in mid-haul logistics fleets.

Chapter "The Electric Location Routing Problem with Time Windows and Partial Recharging" introduces the electric LRP with time windows and partial recharging (ELRP-TWPR) as a fundamental model that considers the location of charging stations and the routing of ECVs simultaneously. Different objective functions are presented minimizing e.g., the total traveled distance, a convex combination of the number of vehicles and charging stations, or total costs. Results are discussed with respect to the impact of these objective functions, analyzing the impact of simultaneous location and routing decisions as well as partial recharging. It is shown that an integrated vehicle routing and charging station location decision improves the objective for any instance of the provided benchmark, whereas improvements due to partial recharging arise only in special cases [2].

Chapter "Are ECVs Breaking Even?—Competitiveness of Electric Commercial Vehicles in Retail Logistics" presents a case study on ECVs in mid-haul retail logistics. It combines aggregated cost analysis with detailed strategic and operational network as well as fleet planning tasks and shows the relevance of such a modeling approach. Herein, it merges total cost of ownership calculations with an LRPIF variant that considers a multi-period, multi-shift planning horizon as well as pickup and delivery characteristics. The study bases on real-world data for the network structure, demand patterns, and vehicle characteristics from the German retail company Tedi. The study shows the applicability of the proposed modeling approach and finds that ECVs are already on the verge of breaking even compared to internal combustion engine vehicles (ICEVs) for the analyzed application case [3, 4].

Chapter "Strategic Planning of Electric Logistics Fleet Networks: A Robust Location-Routing Approach" introduces a robust variant of the ELRP-TWPR using the concept of adjustable robustness. Uncertainties are considered in the customer patterns with respect to spatial distribution, demand, and time windows. To study these uncertainties, benchmark instances are derived based on real-world data, additionally taking different network structures into consideration. In order to assess the significance of considering uncertainties in such a planning task, the results of the robust modeling approach are compared to alternative deterministic modeling approaches (e.g., using average, median, or worst case representations). A parallelized hybrid of adaptive large neighborhood search (ALNS), dynamic programming (DP), and local search (LS) is developed to solve the robust model

using an adversarial approach. The results show the necessity of the robust modeling approach as the deterministic alternatives do not guarantee good or in some cases not even feasible solutions [5].

Chapter "The Impact of Synchronizing Driver Breaks and Recharging Operations for Electric Vehicles" introduces the electric vehicle routing problem with truck driver scheduling (EVRP-TDS) by combining an electric vehicle routing problem with hours of service (HOS) regulations for the European Union (EU) and the United States (US). If recharging on routes and breaking due to HOS regulations is considered simultaneously, a goal conflict arises. This paper shows that this goal conflict can be reduced to a large extent by synchronizing driver breaks and recharging operations. An ALNS with additional DP and request bank components as well as a time efficient evaluation procedure to assess HOS feasibility for both regulations (EU and US) are presented. The analyses focus on the impact of the synchronization potential for the competitiveness of ECVs against ICEVs. Based on benchmark instances derived from real-world data it is shown that the synchronization of charging and breaking due to HOS regulations decreases or even eliminates the disadvantages of ECVs with respect to additional idle times [6].

3 A Generic Algorithmic Framework for LRPIFs and VRPISs

The second part of this thesis contains three articles that develop a comprehensive generic framework representing a new state of the art for the design and operation of logistics networks with intermediate stops. Contrary to other state-of-the-art algorithms that are tailored to specific problem variants, the developed algorithmic framework covers a wide range of application cases and problem variants without need for further tailoring of the algorithm. Besides, a new problem class, the LRPIF, is introduced to provide a formal basis for problems in which the intra-route facility location and the vehicle routing decision are interdependent. The generic algorithmic framework draws a comprehensive frame around both VRPISs and LRPIFs, and thus is unique in providing a new state of the art for a multitude of problem variants from two different problem classes. The developed framework outperforms existing algorithms that are tailored to specific problem variants with respect to solution quality and computational time.

Chapter "An Adaptive Large Neighborhood Search for the Location-Routing Problem with Intra-Route Facilities" introduces a new class of LRPs, the LRPIF, which has high significance for current and future research on efficient and sustainable logistics networks, especially for alternative fuel vehicles and city logistics. Also, it introduces a generic algorithm for the class of LRPIFs. This algorithm bases on an ALNS that is enriched by a corridor guided LS phase and a complex DP component that allows to identify optimal visits to intra-route facilities during the search. Besides developing efficient algorithmic components, this chapter

also focuses on the derivation of complex penalty terms that allow for time efficient move evaluations in O(1). Herein, a new evaluation approach that introduces a corridor-based evaluation of time dependent resources is presented and merged with the concept of time-traveling to avoid the overestimation of feasibility violations on routes. This approach is amenable for arc-based as well as node-based resources. This generic algorithm provides the best known solution quality on the ELRP-TWPR and the battery swap station electric vehicle LRP (BSS-EV-LRP) [7].

Chapter "Designing Sustainable Mid-Haul Logistics Networks with Intra-Route Multi-Resource Facilities" extends the LRPIF and the respective algorithmic framework to combined intra-route facilities that enable the replenishment of multiple resources (e.g., energy and freight) simultaneously. Herein, new penalty terms that allow for time efficient search move evaluations with multiple resources are presented, and an additional lower bounding procedure that prevents the algorithm from searching unpromising facility configurations is derived. The derived algorithm is tested on LRPIF variants with a single resource and yields new best known solutions for both the ELRP-TWPR and the BSS-EV-LRP. The design of experiments and the results discussion focus on the benefit of combined facilities that is shown for the application case of electric logistics fleets. It is shown that allowing for freight replenishment and battery recharging simultaneously leads to cost reductions of up to 45% [8]. These findings are used to create new benchmark instances for the LRPIF with multiple resources.

Chapter "A Solution Framework for a Class of Vehicle Routing Problems with Intermediate Stops" extends the algorithmic framework to a broad class of VRPIS variants. It provides a methodological discussion of the VRPIS, identifies 14 different problem variants, and develops a generic mixed integer problem formulation. To alleviate the algorithmic framework of the previous chapters for these problem variants, time efficient move evaluations and penalty terms as well as resource extension functions for the respective DP component are derived. New best known solutions and the overall best solution quality for four out of five variants that have already been addressed in the literature are provided. For one variant, the second best solution quality is reached [9]. For all variants that have not been addressed in the literature so far, results are stated for further research.

4 Conclusion

Concluding, this thesis provides a profound and generic framework to design and operate LNWIS. Furthermore, it shows the benefit of using this framework for the application case of ECVs in mid-haul logistics. Due to its generality this framework provides a basis amenable for both researchers and practitioners that work in the growing field of LNWIS.

References

1. Schiffer, M., Schneider, M., Laporte, G., Walther, G.: Vehicle routing and location-routing with intermediate stops: a review. Transp. Sci. **53**(2), 319–343 (2019)
2. Schiffer, M., Walther, G.: The electric location routing problem with time windows and partial recharging. Eur. J. Oper. Res. **260**(3), 995–1013 (2017)
3. Schiffer, M., Stütz, S., Walther, G.: Are ECVs breaking even? – Competitiveness of electric commercial vehicles in retail logistics. Les Cahiers du GERAD, G-2017-47 (2017)
4. Schiffer, M., Stütz, S., Walther, G.: Behaviour of Lithium-Ion batteries in electric vehicles. Green energy and technology. In: Electric Commercial Vehicles in Mid-Haul Logistics Networks, 1st edn., pp. 153–173. Springer, Cham (2018)
5. Schiffer, M., Walther, G.: Strategic planning of electric logistics fleet networks: a robust location-routing approach. Omega **80**(1), 31–42 (2017)
6. Schiffer, M., Laporte, G., Schneider, M., Walther, G.: The impact of synchronizing driver breaks and recharging operations for electric vehicles. Les Cahiers du GERAD, G-2017-46 (2017)
7. Schiffer, M., Walther, G.: An adaptive large neighborhood search for the location-routing problem with intra-route facilities. Transport. Sci. **52**(2), 331–352 (2018)
8. Schiffer, M., Schneider, M., Laporte, G.: Designing sustainable mid-haul logistics networks with intra-route multi-resource facilities. Eur. J. Oper. Res. **265**(2), 517–532 (2018)
9. Schiffer, M., Klein, P., Schneider, M., Walther, G.: A solution framework for a class of vehicle routing problems with intermediate stops. Technical report. RWTH Aachen, Aachen (2017)

Exploiting Structure in Non-convex Quadratic Optimization

Jonas Schweiger

Profound knowledge of the mathematical structures is the key to design algorithms that perform practical computations in the most efficient way. A thorough exploitation of such structures is why solver software for so-called linear and mixed-integer linear programs routinely computes optimal solutions for practical problems with hundreds of thousands of variables.

This paper focuses on the contributions to non-convex quadratic programming relaxations from the dissertation [10] for the occasion of the GOR dissertation award 2018. The key results on gas network planning under uncertainty from this dissertation were already presented in this series [9].

Branch-and-bound methods for non-convex quadratic programs (QP) depend on tight relaxations. We contribute in several ways: First, we establish a new way to handle missing linearization variables in the well-known Reformulation-Linearization-Technique (RLT). This is implemented into the commercial software CPLEX. Second, we study the optimization of a quadratic objective over the standard simplex. These basic structure appears as part of many complex models. Exploiting connections to the maximum clique problem and RLT, we derive new valid and strong inequalities. Using exact and heuristic separation methods, we demonstrate the impact of the new inequalities on the relaxation and the global optimization of these problems. Third, we strengthen the state-of-the-art relaxation for the Pooling Problem, a well-known non-convex quadratic problem, which is, for example, relevant in the petrochemical industry. We propose a novel relaxation that captures the essential non-convex structure of the problem but is small enough for an in-depth study. We provide a complete inner description in terms of the extreme points as well as an outer description in terms of inequalities defining its convex hull

J. Schweiger (✉)
Zuse Institute Berlin, Berlin, Germany
e-mail: schweiger@zib.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_7

(which is not a polyhedron). We show that the resulting valid convex inequalities significantly strengthen the standard relaxation of the pooling problem.

1 Reformulations and Relaxations for Quadratic Programs

Quadratic programming deals with the following mathematical program:

$$\begin{aligned} \min_{x \in \mathbb{R}^d} \quad & x^T Q_0 x + a_0 x + b_0 \\ \text{s.t.} \quad & x^T Q_i x + a_i x + b_i \leq 0 && \text{for all } i \in \mathcal{M} \\ & \underline{x_j} \leq x_j \leq \overline{x_j} && \text{for all } j \in \mathcal{N} \end{aligned}$$

The matrices $Q_i \in \mathbb{R}^{d \times d}, i \in \mathcal{M} \cup \{0\}$ are symmetric, but not necessarily positive semidefinite, such that the problem is in general non-convex.

Termwise linearization of all bilinear and quadratic terms is the most common way to convexify quadratic programs. To this end an auxiliary variable Y_{ij} is introduced for each term $x_i x_j$ in the model together with the constraint $Y_{ij} = x_i x_j$. Quadratic terms $Q_{ii} x_i^2$ are over- and underestimated using gradient and secant hyperplanes that might be separated during the solution process. For bilinear terms, the equation $Y_{ij} = x_i x_j$ is non-convex and can be convexified by the so-called *McCormick* inequalities [6]

$$\begin{aligned} \overline{x_j} x_i + \overline{x_i} x_j - \overline{x_i}\,\overline{x_j} \leq Y_{ij}, \qquad & \overline{x_j} x_i + \underline{x_i} x_j - \underline{x_i}\,\overline{x_j} \geq Y_{ij}, \\ \underline{x_j} x_i + \underline{x_i} x_j - \underline{x_i}\,\underline{x_j} \leq Y_{ij}, \qquad & \underline{x_j} x_i + \overline{x_i} x_j - \overline{x_i}\,\underline{x_j} \geq Y_{ij}. \end{aligned}$$

In this fashion all terms appearing in the model are relaxed.

The major drawback of the termwise McCormick relaxation is that each term is relaxed independently, disrespecting the interactions of various constraints. The Reformulation-Linearization Technique (RLT) [11] is a way to combine several aspects of the model and derive new valid inequalities.

RLT consists of two steps. In the first step, valid constraints are multiplied by other constraints or by variables yielding an equation or inequality with higher order terms. In the second step, these terms are reformulated using linearization variables to obtain a linear constraint. The result is a valid constraint on the linearization variables that is often a very strong cutting plane [7].

Our contribution is to propose a treatment for those quadratic terms that arise from RLT but do not appear in the model so far. To avoid the introduction of new linearization variables and still be able to formulate a valid inequality, we propose to project them out by replacing them with appropriate over- and underestimators, an approach that has not been described or applied before to the best of our knowledge. This approach is implemented in the commercial solver CPLEX 12.7.0 and enabled

by default for problems with non-convex quadratic objective. On the CPLEX test set at IBM comprising several hundred problems, projected RLT is instrumental for solving eleven additional instances in comparison to version 12.6.3. When looking at models that are affected by the separation of theses inequalities, a run time reduction of 29% is achieved. This rises to 84% when focusing on "hard instances" where either of the two versions takes at least 1000 s to solve the problem. This work on RLT and the work of the next section is joint work with Pierre Bonami, Andrea Lodi, and Andrea Tramontani [2].

2 Motzkin-Straus Inequalities for Standard Quadratic Programming and Generalizations

Next, we study more specific structures that appear as part of many complex models. The *standard simplex* is the set $\Delta = \left\{ x \in \mathbb{R}^d \;\middle|\; \sum_{i=1}^{d} x_i = 1, x \geq 0 \right\}$. We study the optimization of a quadratic function over the standard simplex; a problem called *Standard Quadratic Program* [1]:

$$\min \left\{ x^T Q x \;\middle|\; x \in \Delta \right\}. \tag{StQP}$$

The next theorem connects the clique number of a graph with (StQP).

Theorem 1 (Motzkin-Straus [8]) *Let A be the adjacency matrix of a simple, undirected graph G with d nodes and $\omega(G)$ its clique number. Then, the following relation holds:*

$$\max \left\{ x^T A x \;\middle|\; x \in \Delta \right\} = 1 - \frac{1}{\omega(G)}.$$

Again, we assume the quadratic function has been reformulated with auxiliary linearization variables Y and we strive to use Theorem 1 to derive strong valid inequalities for the set

$$\Gamma = \left\{ (x, Y) \in \mathbb{R}^d \times (\mathbb{R}^d \times \mathbb{R}^d) \;\middle|\; Y = x x^T, x \in \Delta \right\}.$$

Corollary 1 *For any simple, undirected graph G with adjacency matrix A and clique number $\omega(G)$, the following inequality is valid for $(x, Y) \in \Gamma$:*

$$\langle A, Y \rangle \leq 1 - \frac{1}{\omega(G)}$$

In the remainder, we call the inequalities derived from Corollary 1 *Motzkin-Straus Clique inequalities (MSC inequalities)*.

We propose several heuristic methods to separate violated MSC inequalities. For complete bipartite graphs, the separation can be formulated as a binary QP which gives an exact separation algorithm for this class of inequalities. We show that the relaxation is considerably strengthened and that the separation program for complete bipartite graphs can be solved very efficiently by state-of-the-art solvers.

By performing a specific aggregation of RLT inequalities, we can obtain a nonlinear inequality that generalizes Motzkin-Straus Clique inequalities for bipartite graphs. Consider a partition $(M, \bar{M})$ of $\{1, \ldots, d\}$. First, we multiply the simplex constraint $\sum_{i=1}^{d} x_i = 1$ by some x_j and add these equation up for all $j \in M$. Then, we subtract the term $x_i x_j$ for $i, j \in M$ from both side and get

$$\sum_{j \in M} \sum_{i \in \bar{M}} x_i x_j = \sum_{j \in M} x_j - \sum_{j \in M} \sum_{i \in M} x_i x_j.$$

Formulating the left hand side with the linearization variables and noting that the right hand side can be overestimated by tangents on the function $g = z - z^2$ yields the following

Theorem 2 *Let f_α be the tangent of $g = z - z^2$ at $\alpha \in [0, 1]$. Then, for any $M \subset \{1, \ldots, d\}$, the following inequality is valid for $(x, Y) \in \Gamma$:*

$$\sum_{j \in M} \sum_{i \in \bar{M}} Y_{ij} \leq f_\alpha \left(\sum_{j \in M} x_j \right)$$

We call these inequalities *generalized MSC bipartite inequalities.* They can also be separated by solving a quadratic programming problem which features continuous and binary variables. An extensive computational study shows that the inequalities considerably strengthen the McCormick relaxation and yield a significant improvement in terms of dual bound and time to optimality. On a large test set, these inequalities close over 86% of the remaining gap of the root relaxation including the application of RLT. With reference to CPLEX version 12.6.3, combining the strengths of projected RLT and both classes of inequalities, we increased the number of instances solved from 119 to 230. The run time as well as the number of branch-and-bound nodes fall by one or two orders of magnitude depending on the problem size. The proposed inequalities can be adapted to sets described by a knapsack constraint with more general coefficients.

3 Strong Relaxations for the Pooling Problem

The pooling problem is a classic non-convex, nonlinear problem introduced by Haverly in 1978 [3], which is, for example, relevant in the petrochemical industry. The task is to route flow from inputs over pools to outputs. Thematerial has attributes

whose concentrations are known at inputs, but are constrained at outputs. The challenge is to route the flow through the network such that the quality constraints at the outputs are met. The blending of the material at pools and outputs differentiates the pooling problem from other multi-commodity flow problems and give rise to non-convex quadratic constraints. This part is joint work with Jim Luedtke, Claudia D'Ambrosio, and Jeff Linderoth [4, 5].

We use the state-of-the-art *pq*-formulation [12] as starting point to derive a novel relaxation for the problem. This is done by reducing the network to its minimal essence comprising only an aggregated version of the inputs, one output, one pool and one material attribute and finally to the study of the feasible set for the following constraints:

$$u - xt = 0 \tag{1a}$$

$$y + u \leq 0 \tag{1b}$$

$$z + x \leq C \tag{1c}$$

$$\underline{\beta} z \leq y \leq \overline{\beta} z \tag{1d}$$

$$z \geq 0,\ x \in [0, C],\ t \in [\underline{\gamma}, \overline{\gamma}] \tag{1e}$$

All variables are linear aggregations of variables from the *pq*-formulation and the set is carefully constructed to be a relaxation. Thus valid inequalities for the set are also valid for the *pq*-formulation. This set is small enough for a thorough study, but still captures parts of the central non-convex structure of the pooling problem in the non-convex constraint (1a).

After finding the relaxation, we follow the route that has been walked by integer linear programmers for decades, namely to study the extreme points of the relaxation and translate the inner description of the set into an outer description based on the defining inequalities. The set of interest is in general not a polyhedron and has infinitely many extreme points. However, a finite subset of the extreme points leads to a parametrized set of linear inequalities that are used to derive new valid linear and convex nonlinear inequalities for the pooling problem. The inequalities are added directly or by means of the separation of gradient inequalities to the *pq*-formulation. A computational study shows that the proposed relaxation is tight enough to strengthen the McCormick relaxation of the *pq*-formulation. On a test set of 360 instances the gap of the state-of-the-art *pq*-relaxation is reduced from 5.5 to 3.0%. Especially on sparse instances, the additional inequalities provide a significant speed-up of the global solution of the problem and allow to solve several instances that are not solved by the *pq*-formulation.

4 Conclusion

We focused on applications, where nonconvexity in the model formulation pose principal difficulty and improved the solution methods by exploiting structural particularities. Different techniques were used to achieve these results. For RLT we propose a small enhancement for a well-known general technique. For standard quadratic programming, we use a connection to a combinatorial problem to derive cutting planes that were then generalized using a connection to RLT. In contrast, for the pooling problem, the key contribution was to find and study an appropriate relaxation to derive valid inequalities. We conducted extensive computational experiments in order to fathom the computational impact.

Acknowledgements The author thanks Pierre Bonami, Claudia D'Ambrosio, Jeff Linderoth, Andrea Lodi, Jim Luedtke, and Andrea Tramontani for the joint work.

References

1. Bomze, I.M.: On standard quadratic optimization problems. J. Glob. Optim. **13**(4), 369–387 (1998)
2. Bonami, P., Lodi, A., Schweiger, J., Tramontani, A.: Solving quadratic programming by cutting planes. SIAM J. Optim. **29**(2), 1076–1105 (2018). https://doi.org/10.1137/16M107428X
3. Haverly, C.A.: Studies of the behavior of the recursion for the pooling problem. SIGMAP Bullet. **25**, 19–28 (1978)
4. Luedtke, J., D'Ambrosio, C., Linderoth, J., Schweiger, J.: Strong Convex Nonlinear Relaxations of the Pooling Problem. Technical report ZIBReport 18-12. Zuse Institute, Berlin (2018)
5. Luedtke, J., D'Ambrosio, C., Linderoth, J., Schweiger, J.: Strong Convex Nonlinear Relaxations of the Pooling Problem: Extreme Points. Technical report ZIB-Report 18-13. Zuse Institute, Berlin (2018)
6. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I convex underestimating problems. Math. Program. **10**, 147–175 (1976)
7. Misener, R., Floudas, C.A.: GloMIQO: Global mixed-integer quadratic optimizer. J. Glob. Optim. **57**(1), 3–50 (2013)
8. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turán. Canadian J. Math. **17**, 533–540 (1965)
9. Schweiger, J.: Gas network extension planning for multiple demand scenarios. In: Lübbecke, M., Koster, A., Letmathe, P., Madlener, R., Peis, B., Walther, G. (Eds.) Operations Research Proceedings 2014: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR), RWTH Aachen University, Germany, Sept 2–5, 2014, pp. 539–544. Springer International Publishing, Cham (2016)
10. Schweiger, J.: Exploiting structure in nonconvex quadratic optimization and gas network planning under uncertainty. PhD thesis. Technische Universität, Berlin (2017)
11. Sherali, H.D., Alameddine, A.R.: A new reformulation-linearization technique for bilinear programming problems. J. Glob. Optim. **2**(4), 379–410 (1992)
12. Tawarmalani, M., Sahinidis, N.V.: Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications. Kluwer Academic Publishers, Boston (2002)

Part II
Business Analytics, Artificial Intelligence and Forecasting

Military Manpower Planning Using a Career Path Approach Applied to the Belgian Defense

Oussama Mazari Abdessameud, Filip Van Utterbeeck, and Marie-Anne Guerry

1 Introduction

The military organization is composed of a number of job positions which have to be occupied by soldiers. Certain characteristics are required for each job position. The military organization specificity is having a strict hierarchical structure. The personnel movement in the organization is restricted by some internal mobility rules. Also the recruitment is limited to the lowest rank [1, 2]. Due to these strict rules, military manpower planning is very important. The military manpower planning aims to find the adequate human resources management policies to reach and maintain the required military manpower.

Wang describes effective military manpower planning as "there will continue to be sufficient people with the required competencies to deliver the capability output required by the Government at affordable cost"[3]. To be able to tackle a military manpower planning problem, a description of the specificity of the military manpower is needed.

The military organization needs personnel to occupy a number of job positions. These job positions require that the occupant has specific characteristics. The most used characteristics are military ranks, affiliations and individual competences. Each of these characteristics can change or evolve through time. For example, the military rank advances with promotion, and competences are improved or new ones are gained through training.

O. Mazari Abdessameud (✉) · F. Van Utterbeeck
Department of Mathematics, Royal Military Academy, Brussels, Belgium
e-mail: mazari.abdessameud.oussama@gmail.com

M.-A. Guerry
Department of Business Technology and Operations, Vrije Universiteit Brussel, Brussels, Belgium

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_8

Every soldier in the organization is recruited as a trainee. He undergoes a training to gain a basic competence in the military organization, which allows him to fulfill a job position. As his seniority increases, he is likely to get rank promotions and become eligible for other job positions. A soldier can also follow advanced training throughout his career to meet the requirements of job positions with higher responsibilities. The soldiers' career can terminate by a retirement, which corresponds to a natural attrition at the end of a career. In the case of unnatural attrition [4], the soldier leaves the organization under other circumstances such as medical issues.

Military manpower planning consists of two interdependent logics. It relies on statutory logic and competence logic. The statutory logic considers the manpower evolution on the strategic level. The strategic goal is attainability and/or maintainability [5]. The attainability concerns reaching a certain statutory manpower distribution. The maintainability is getting a steady state manpower distribution where we can ensure that the manpower is maintainable [6]. The statutory logic tackles the recruitment, promotion and retirement policies in order to meet the targeted goals. On the other hand, the competence logic targets the assignment of the right person with the required characteristics to a suitable position to get the required competence manpower distribution.

We define a manpower distribution as the number of personnel in each subgroup following a certain classification of the manpower. If the considered classification uses a strategic logic classification of the manpower (such as rank) we call it statutory manpower distribution. On the other hand, if it uses a competence logic classification (such as job type) we call it competence manpower distribution.

In previous work, the statutory logic is mainly addressed with system dynamics or Markov chain models. An et al. [7] present a review about the use of system dynamics for the statutory logic of manpower planning. Also, Guerry and De Feyter [8] present a review on the use of Markov chains in the manpower context for the statutory logic. The competence logic is mostly tackled using optimization techniques. Cai et al. [9] approach manpower allocation problems using a minimum cost flow model. Hall and Fu [10] rely on a network flow model combined with linear programming to find an optimal competence manpower distribution.

However, these two defined logics are interdependent and affect each other. Therefore, considering both logics simultaneously would result in a more elaborate approach. Gass [11] uses a Markov model to approach the statutory logic which is coupled with a network flow to resolve the competence logic problem. Mazari Abdessameud et al. [12] proposes a simultaneous optimization of the two logics based on a flow network model combined with a goal programming approach to find the optimal way to reach the aimed statutory and competence distributions.

The possibilities for the evolution in a career within a military organization can be captured by a rather limited number of career paths. This motivates introducing a career path approach. In this article, we present a career path model allowing the consideration of both logics at the same time. Our model is applied to the Belgian defense manpower to find optimal policies in order to reach the demanded manpower.

2 Career Path Modeling Approach

During his service in the military organization, each soldier is characterized by his characteristics state. The characteristics state of a soldier gathers the information regarding his characteristics (job position, rank, acquired skills and competences, etc...). The evolution through time of this characteristics state is called the soldier's career path [13]. Having a limited number of characteristics makes the number of possible career paths in the military organization limited. Moreover, not all feasible career paths have to be considered as certain career paths might be undesirable either from the point of view of the organization or the point of view of the soldier.

Baumgarten defines in his dissertation [14] a model named "Career path selection". This model aims to determine the degree to which the organization's requirement can be fulfilled by developing qualified military personnel and to assign the developed manpower into suitable career paths. The established model considers the possible career paths and focuses only on the assignment of each soldier to a suitable career path. However, the manpower initially available in the organization was not considered in this work.

2.1 Career Path Model Construction

The career path model consists of representing the military organization using career paths. The first step is the identification of possible career paths. The military manpower is then assigned to the different career paths. Every military organization has a limited number of characteristics that could be attributed to the soldiers. Thus, for each military organization we have a limited number of possible characteristics states, which results in a limited number of career paths. However, the number of the generated career paths could be enormous and causes a heavy burden to handle. Fortunately, only the career paths that obey the human resources management policies defined by the organization have to be retained. For example, if the organization policy states that a minimum stay in a rank is 5 years, all career paths with a stay within a rank less than 5 years are ignored.

2.2 Optimal Solution Construction

Representing the organization with a career path approach means that the manpower in the organization is divided into subsets according to the identified possible career paths. If we suppose that the organization has a known capacity of recruitment each year, the operational goal of the organization as well as the strategic one are translated into finding the optimal amounts to recruit in each career path.

The recruited amounts in each career path are fractions of the recruitment capacity of the organization. In order to find the optimal fractions, we resort to mathematical programming. An important aspect to take into consideration is the fact that the available workforce is not always equal to the demand. Having this deviation between the required and the available manpower, goal programming is an appropriate approach to use.

To write the goal program, we consider three types of variables: On the one hand the number of recruited soldiers that are assigned to each career path. On the other hand the number of the initial employees in each of the career paths, and finally the deviations. The deviation variables are used in the soft constraints to define the deviation from the targeted goal.

3 Career Path Approach Applied to the Belgian Defense

We applied the career path approach to the Belgian defense. In order to define the characteristics state, we considered for this application some specific characteristics which are: the frame, the career type, the affiliation and the job position type. The frame is defined by the academic level of the soldier and his rank class. The frame has five possible values, two values for the officers' class, two for the noncommissioned officers' class and one for the enlisted personnel. The career type expresses whether the soldier is engaged for short-term or he will continue until retirement. The affiliation defines the main component of the soldiers. In our application, we consider 16 affiliations including: aviation, marine and infantry. The job position type, which has three possible values, defines if the soldier is under training, working on his first occupation in the organization or working on advanced occupation.

Based on the defined characteristics state and the human resource management directorate of the Belgian defense information, we identify 432 possible career paths. We received a personnel database sized 27,700 current soldiers. The goal is to reach a population of 25,000 as soon as possible with certain distributions on the statutory level and the competence level. As for the distribution over the different affiliations, we want to have an equal distribution. The other targeted distributions will be shown with the results.

For a simulation of 25 years, the generated goal program is sized at 26,752 variables and 1073 constraints. We use a CPLEX solver to solve the problem. The following figures are derived form the obtained results. They represent the evolution of the distributions based on each considered characteristic. The targeted distributions are shown in dotted lines.

Figure 1 shows the evolution of the total manpower through time. We notice that we could reach the targeted population size within 7 years. Figure 2 illustrates the statutory distributions based on the frame and the career type. We notice that both statutory distributions are fulfilled by the seventh year. The competence distributions are shown in Fig. 3. The affiliation distribution in Fig. 3a is fulfilled within 7 years. We have equal manpower in each affiliation. However, the distribution based on the job type (Fig. 3b) is not fully satisfied. This is due to the restrictions made by the career paths reflecting the organization's policies which do not allow the targeted distribution to take place.

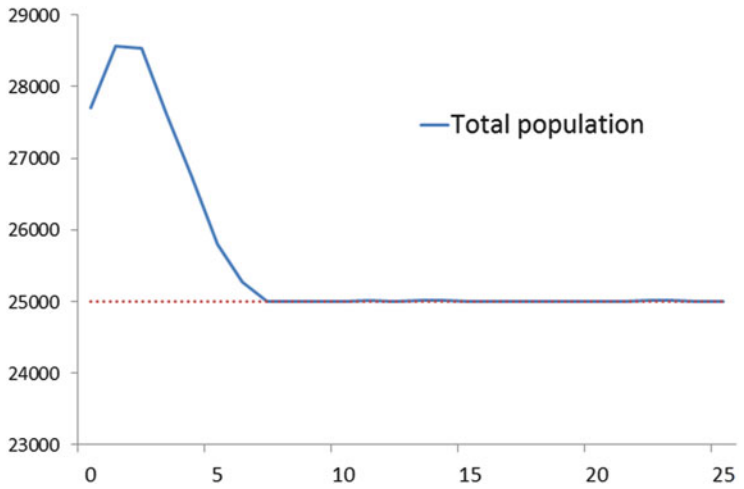


Fig. 1 Total population evolution

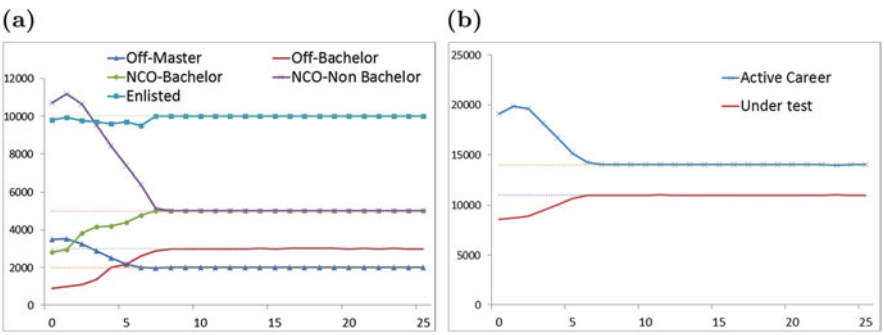


Fig. 2 Statutory distributions of the manpower. (**a**) Distribution based on the frame. (**b**) Distribution based on the career type

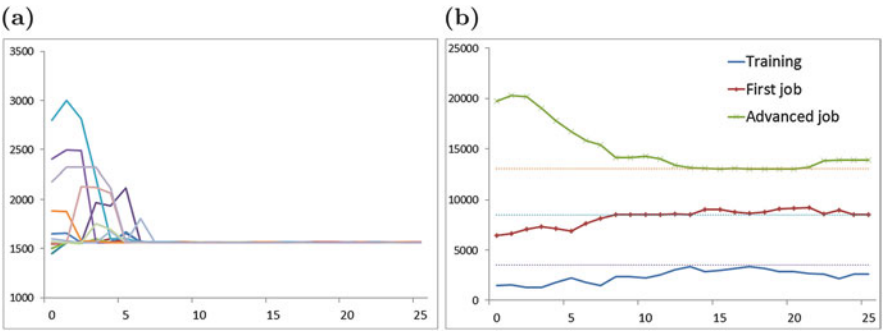


Fig. 3 Competence distributions of the manpower. (**a**) Distribution based on the affiliation. (**b**) Distribution based on the job type

4 Conclusions

The proposed model is able to propose a solution to our problem. The solution is an assignment plan for the available manpower and a recruitment plan for the coming years. These plans allow the military organization to reach the targeted manpower distribution on both the statutory level and the competence one. The strength of this model is the ability to adapt career paths to the organization's policies, also the simultaneous optimization of statutory and competence logics.

References

1. Jaquette, D.L., Nelson, G.R., Smith, R.J.: An analytic review of personnel models in the Department of Defense, R-1920-ARPA . RAND Corporation, Santa Monica, CA (1977). As of April 23, 2019: https://www.rand.org/pubs/reports/R1920.html
2. Hall, A.O.: Simulating and optimizing: military manpower modeling and mountain range options. Thesis (2009). https://drum.lib.umd.edu/handle/1903/9299
3. Henry, T., Ravindran, R.: A goal programming application for army officer accession planning. Inf. Syst. Oper. Res. **43**, 111–119 (2005)
4. Setlhare, K.: Optimization and Estimation Study of Manpower Planning Models. University of Pretoria, Pretoria (2007)
5. Bartholomew, D.J.: Manpower Planning. Encyclopedia of Operations Research and Management Science, pp. 910–912. Springer, New York (2013)
6. Akaninyene, U.U., Peter, I.U.: Optimal maintainability of manpower system with time invariant coefficients. J. Stat. Manag. Syst. **21**, 455–466 (2018)
7. An, L., Jeng, J.J., Lee, Y.M., Ren, C.: Effective workforce lifecycle management via system dynamics modeling and simulation. In: Proceedings - Winter Simulation Conference, pp. 2187–2195 (2007)
8. Guerry, M.-A., De Feyter, T.: Markovian approaches in modeling workforce systems. J. Curr. Issues Financ. Bus. Econ. **2**, 351–370 (2009)

9. Cai, Y., Zhang, Z., Guo, S., Qin, H., Lim, A.: A tree-based tabu search algorithm for the manpower allocation problem with time windows and job-teaming constraints. In: International Joint Conferences on Artificial Intelligence, pp. 496–502 (2013)
10. Hall, A.O., Fu, M.C.: Optimal army officer force profiles. Optim. Lett. **9**, 1769–1785 (2015)
11. Gass, S.I.: Military manpower planning models. Comput. Oper. Res. **18**, 65–73 (1991)
12. Mazari Abdessameud, O., Van Utterbeeck, F., Van Kerckhoven, J., Guerry, M.-A.: Military manpower planning - towards simultaneous optimization of statutory and competence logics using population based approaches. In: ICORES International Conference on Operations Research and Enterprise Systems (2018)
13. Baruch, Y.: Transforming careers: from linear to multidirectional career paths: organizational and individual perspectives. Career Dev. Int. **9**, 58–73 (2004)
14. Baumgarten, P.B: Optimization of United States Marine Corps Officer Career Path Selection. Thesis (2000). https://calhoun.nps.edu/handle/10945/32952

Profit-Oriented Feature Selection in Credit Scoring Applications

Nikita Kozodoi, Stefan Lessmann, Bart Baesens, and Konstantinos Papakonstantinou

1 Introduction

One of the most important tasks in credit risk analytics is to decide upon loan provisioning. Binary scoring systems are widely deployed to support decision-making and predict applicants' willingness and ability to repay debt. Financial institutions face costs of gathering and storing large amounts of data on customer behavior used to score applicants. In addition, companies need to comply with regulation that enforces comprehensible models. Feature selection aims at solving this problem by removing irrelevant data, which can reduce costs and improve the scorecard performance and interpretability.

Recent literature criticized a widespread practice of using standard performance measures such as area under the receiver operating characteristic curve (AUC) for evaluating scoring models [7]. Relying on profit-based indicators may improve scorecard profitability [5, 15]. This finding stresses the importance of using value-oriented feature selection strategies that identify the optimal subset of variables

N. Kozodoi (✉)
Humboldt University of Berlin, Berlin, Germany

Kreditech, Hamburg, Germany
e-mail: nikita.kozodoi@hu-berlin.de

S. Lessmann
Humboldt University of Berlin, Berlin, Germany

B. Baesens
Catholic University of Leuven, Leuven, Belgium

K. Papakonstantinou
Kreditech, Hamburg, Germany

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_9

in a profit-maximizing manner. The goal of this paper is to introduce the profit maximization framework to the feature selection stage to facilitate the business-driven model development.

We develop a wrapper-based feature selection framework that uses the Expected Maximum Profit measure (EMP) as a fitness function. EMP has been previously used in credit scoring for model evaluation [15]. The advantage of the proposed approach is that it searches for variable subsets that optimize the business-inspired indicator. To validate the effectiveness of our method, we conduct an empirical experiment on multiple credit scoring data sets.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on profit-driven credit scoring and feature selection methods. Section 3 describes our experimental setup, whereas Sect. 4 presents empirical results. In Sect. 5, we discuss the main conclusions of our study.

2 Related Literature

2.1 Profit-Oriented Credit Scoring

The credit scoring literature has proposed several profit measures to improve the quality of scorecards. Serranco-Cinca et al. use the internal rate of return based on the loan interest [12]. Finlay proposes estimating a contribution of each applicant to the profit of the financial institution [5]. Both these measures imply replacing binary default indicator by a continuous target variable and therefore transform a classification problem into the regression task.

Recently, Verbraken et al. developed the EMP measure [15]. EMP is based on the costs of the incorrect classification of *bad* loans (defaults) and benefits of the correct prediction of *good* ones (repayers). It can be computed as:

$$\text{EMP} = \int_0^1 \Big[B \cdot \pi_0 F_0(t) - C \cdot \pi_1 F_1(t) \Big] f(B) d(B), \tag{1}$$

where B is the expected loss in case of default and C is the return on the investment, π_i are prior probabilities of *good* and *bad* loans, and $F_i(t)$ are predicted cumulative fractions of class i based on cutoff t. The return on investment is assumed to be constant, whereas the expected loss is a stochastic variable based on the loss given default and exposure at default (see [15] for details).

EMP can be interpreted as the incremental profit from deciding on credit applications using a scorecard compared to a baseline scenario where credits are granted without screening. In this paper, we use the EMP criterion to measure the scorecard profitability.

2.2 Feature Selection

Feature selection methods split into filters, wrappers and embedded methods [6]. Filters rank and select features based on some general data characteristics. Popular measures include correlation, information gain and others [4]. Filters are fast and efficient but they were shown to perform poorly compared to wrappers and embedded methods [6]. Embedded methods conduct feature selection simultaneously with the model training. One of the popular approaches is recursive feature selection using the SVM framework [4]. The drawback of embedded methods is that they can only be applied within a specific model.

Wrappers go through different feature subsets and select the optimal subset based on the model performance. Since evaluating all possible feature combinations is computationally expensive, research has suggested heuristic search strategies. Popular approaches are sequential forward selection (SFS) and sequential backward selection (SBS) [6]. SFS starts with an empty model and iteratively adds features, selecting the one which brings the largest performance gain, whereas SBS starts with a full set of features and eliminates those contributing the least to the model performance. The search is continued until there is no further improvement. Another strategy relies on evolutionary algorithms such as genetic algorithms (GA) [16]. GAs operate on a population of individuals, where each individual represents a model with binary genes indicating inclusion of specific features. At each generation, a new population is created by selecting individuals according to their fitness (model performance), recombining them together and undergoing mutation. The individual with the highest fitness is selected after running multiple generations.

The literature on profit-oriented credit scoring focuses on model selection and parameter estimation but does not consider the feature selection stage. Existing studies on value-driven feature selection focus on feature costs. Some researchers suggest using a budget constraint that limits the maximal cost of the selected features [11]. Another approach is to use cost-adjusted ranking criteria when applying filter methods [3].

To the best of our knowledge, research on value-driven feature selection in credit scoring is currently limited to the embedded regularization framework for SVM [9, 10]. Recent benchmarking studies in credit scoring have shown that SVM performs poorly in comparison with other classifiers [8]. Given these results, developing a profit-driven feature selection approach that is not limited to SVM contributes to the literature. In this paper, we focus on wrappers due to their flexibility and better performance compared to filters.

3 Experimental Setup

3.1 Data Sets

The empirical evaluations are based on ten retail credit scoring data sets. Data sets *australian* and *german* stem from the UCI Repository [14]. The data sets *pakdd*, *lendingclub* and *gmsc* were provided by different companies for the data mining competitions on PAKDD and Kaggle platforms. Datasets *bene1*, *bene2* and *uk* were collected from financial institutions in the Benelux and UK [1]. The *thomas* data set is provided by Thomas et al. [13], whereas *hmeq* was collected by Baesens et al. [2].

Each of the data sets has a unique set of features describing the loan applicant (e.g., gender, income) and loan characteristics (e.g., amount, duration). Some data sets also include information on previous loans of the applicant. The target variable is a binary indicator whether the customer has repaid the loan or not. Table 1 summarizes the main characteristics of the data sets.

3.2 Modeling Framework

The modeling pipeline consists of several stages. First, each data set is preprocessed in the same way. We impute missing values with means for continuous features and with modes for categorical features. Next, we encode categorical variables with $k-1$ dummies, where k is the number of unique categories.

The data sets are randomly partitioned into two subsets: training (70% cases) and holdout data (30%). On the training set, we use fivefold cross-validation to perform feature selection. Then, we use the whole training set to train classification models with the identified feature subsets and evaluate results on the holdout data. The partitioning is repeated ten times on each data set.

On a feature selection stage, we use three wrappers: SFS, SBS and GA. The parameters of GA were selected based on the grid search on a subset of training

Table 1 Credit scoring data sets (after preprocessing)

Data label	Sample size	Num. features	Default rate
australian	690	42	0.4449
german	1000	61	0.3000
thomas	1225	28	0.2637
bene1	3123	83	0.3333
hmeq	5960	20	0.1995
bene2	7190	28	0.3000
uk	30,000	51	0.0400
lendingclub	43,344	206	0.1351
pakdd	50,000	373	0.2608
gmsc	150,000	68	0.0668

data: number of generations and number of individuals were set to 200. For each of the feature selection algorithms, we use two performance measures as objective functions. Relying on EMP as a fitness function is a central element of our approach, whereas using AUC serves as a benchmark for the standard feature selection techniques. Logistic regression is used as a base classifier.

4 Empirical Results

The performance of different methods is compared by the mean model ranks in terms of AUC and EMP. To compute the ranks, we order all algorithms by AUC and EMP values within each modeling trial on each of the data sets, and average the model positions. The results are depicted in Fig. 1.

For all three wrappers, EMP optimization during feature selection generalizes to a higher expected profit on the new data. Therefore, our results emphasize the importance of selecting the appropriate fitness function in the early stages of scorecard development. If the goal of the scoring model is to maximize the expected profit, this measure should also be used as an objective for feature selection. Relying on AUC as one of the standard performance measures results in a suboptimal scorecard with a lower EMP.

We also observe differences in terms of the number of selected features. For the sequential methods, EMP-driven wrappers reach the stopping criteria earlier, resulting in a lower average number of the selected features for SFS (14 compared

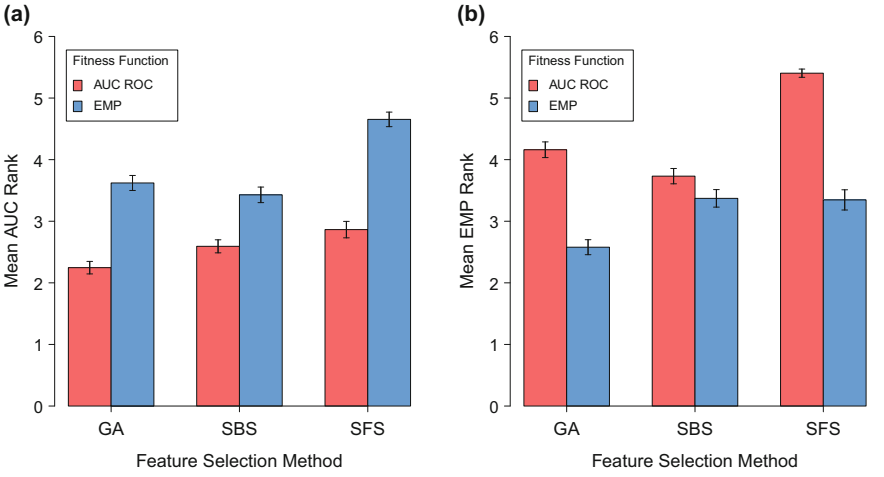


Fig. 1 Mean ranks of different feature selection methods in terms of (**a**) AUC (left) and (**b**) EMP (right). The ranks are computed for six methods and aggregated across ten trials on ten data sets. Whiskers refer to intervals of one standard error around the mean

to 25) and a higher number of features for SBS (89 compared to 81). These results provide evidence that using a profit-driven fitness function may also lead to a faster convergence and help reducing the computational time.

5 Conclusions and Future Work

This paper presents a profit-driven framework for feature selection in credit scoring. We use the recently developed EMP measure as a fitness function for wrapper-based feature selection. The effectiveness of our approach is evaluated on ten real-world retail credit scoring data sets.

Empirical results indicate that the proposed profit-maximizing feature selection framework identifies variable subsets that yield a higher expected profit compared to methods based on standard performance measures. These results stress the importance of using the business-inspired metrics for feature selection. Relying on a standard practice of using statistical measures such as AUC may lead to scorecards with a lower profitability, which motivates implementing the profit maximization on different stages of the model development.

Future research could pursue several directions. For practitioners, it would be important to extend the profit-driven framework to other stages of model development. A benchmarking study with a rich set of EMP-based wrappers would help identifying the optimal search strategy for profit-driven feature selection. Another direction would be to use the developed approach in other business applications such as customer churn.

References

1. Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., Vanthienen, J.: Benchmarking state-of-the-art classification algorithms for credit scoring. J. Oper. Res. Soc. **54**(6), 627–635 (2003). https://doi.org/10.1057/palgrave.jors.2601545
2. Baesens, B., Roesch, D., Scheule, H.: Credit Risk Analytics: Measurement Techniques, Applications, and Examples in SAS. Wiley, Hoboken (2016). https://dx.doi.org/10.1002/9781119449560
3. Bolón-Canedo, V., Sánchez-Maroño, N., Alonso-Betanzos, A.: Recent advances and emerging challenges of feature selection in the context of big data. Knowl.-Based Syst. **86**, 33–45 (2015). https://dx.doi.org/10.1016/j.knosys.2015.05.014
4. Chandrashekar, G., Sahin, F.: A survey on feature selection methods. Comput. Electr. Eng. **40**(1), 16–28 (2014). https://dx.doi.org/10.1016/j.compeleceng.2013.11.024
5. Finlay, S.: Credit scoring for profitability objectives. Eur. J. Oper. Res. **202**(2), 528–537 (2010). https://dx.doi.org/10.1016/j.ejor.2009.05.025
6. Guyon, I., Gunn, S., Nikravesh, M., Zadeh, L.A.: Feature Extraction: Foundations and Applications. Studies in Fuzziness and Soft Computing. Springer, Berlin (2006). https://dx.doi.org/10.1007/978-3-540-35488-8
7. Hand, D.J.: Good practice in retail credit scorecard assessment. J. Oper. Res. Soc. **56**(9), 1109–1117 (2005). https://dx.doi.org/10.1057/palgrave.jors.2601932

8. Lessmann, S., Baesens, B., Seow, H.V., Thomas, L.C.: Benchmarking state-of-the-art classification algorithms for credit scoring: an update of research. Eur. J. Oper. Res. **247**(1), 124–136 (2015). https://dx.doi.org/10.1016/j.ejor.2015.05.030
9. Maldonado, S., Bravo, C., Lopez, J., Pérez, J.: Integrated framework for profit-based feature selection and SVM classification in credit scoring. Decis. Support. Syst. **104**, 113–121 (2017). https://doi.org/10.1016/j.dss.2017.10.007
10. Maldonado, S., Pérez, J., Bravo, C.: Cost-based feature selection for support vector machines: an application in credit scoring. Eur. J. Oper. Res. **261**(2), 656–665 (2017). https://dx.doi.org/10.1016/j.ejor.2017.02.037
11. Min, F., Hu, Q., Zhu, W.: Feature selection with test cost constraint. Int. J. Approx. Reason. **55**(1), 167–179 (2014). https://dx.doi.org/10.1016/j.ijar.2013.04.003
12. Serrano-Cinca, C., Gutiérrez-Nieto, B: The use of profit scoring as an alternative to credit scoring systems in peer-to-peer (P2P) lending. Decis. Support. Syst. **8**, 113–122 (2016). https://dx.doi.org/10.1016/j.dss.2016.06.014
13. Thomas, L.C., Edelman, D.B., Crook, J.N.: Credit Scoring and Its Applications. SIAM, Philadelphia (2002). https:dx.doi.org/10.1137/1.9780898718317
14. UCI Machine Learning Repository. http://archive.ics.uci.edu/ml/
15. Verbraken, T., Bravo, C., Weber, R., Baesens, B.: Development and application of consumer credit scoring models using profit-based classification measures. Eur. J. Oper. Res. **238**(2), 505–513 (2014). https:dx.doi.org/10.1016/j.ejor.2014.04.001
16. Yang, J., Honavar, V.: Feature subset selection using a genetic algorithm. In: Feature Extraction, Construction and Selection, pp. 117–136. Springer, Boston (1998)

A Maturity Model for the Classification of Real World Applications of Data Analytics in the Manufacturing Environment

Thomas Pschybilla, Daniel Baumann, Wolf Wenger, Dirk Wagner, Stephan Manz, and Thomas Bauernhansl

1 Introduction

The progressing digital transformation in manufacturing environment leads to an extensive interconnection of entire supply chain networks. This leads to an increase of available data but not necessarily to an increase of knowledge since knowledge and data are not equivalent [1]. The usage of data analytics enables companies to gain relevant insights from production data and support their decision-making processes [2]. Although the importance of data analytics for manufacturing is undisputed, companies are still in an early stage [3].

This article provides an introduction into the topic of data analytics in manufacturing. A theory-driven maturity model will be presented that can be used by companies to assess their use cases in the context of manufacturing and data analytics. It enables the user to evaluate the current maturity level and identify potentials for further developments. The model also aims to subdivide the vast and complex topic area and match relevant tools in data analytics.

The field of data mining can be understood as an advanced spectrum of methods which enables a predominately automated acquisition of knowledge from large amounts of data [4]. Knowledge describes the cross-linking of information with

T. Pschybilla (✉) · D. Baumann · D. Wagner · S. Manz
TRUMPF GmbH + Co. KG, Ditzingen, Germany
e-mail: thomas.pschybilla@trumpf.com

W. Wenger
Baden-Württemberg Cooperative State University, Stuttgart, Germany
e-mail: wolf.wenger@dhbw-stuttgart.de

T. Bauernhansl
IFF, University of Stuttgart, Stuttgart, Germany
e-mail: thomas.bauernhansl@iff.uni-stuttgart.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_10

regard to a specific issue and is the basis for making decisions and putting them into action [1, 5].

2 Development of the Method

Maturity models are a significant tool to identify the current maturity level and possible directions for further developments. A maturity model can be divided not only into maturity levels but also thematically into different dimensions [6, 7].

2.1 Maturity Models

This article presents a grid-based maturity model. This form of model enables the user to allocate the maturity of use cases based on descriptions provided for every level on different dimensions [6]. Maturity grids enable an efficient and fast self-assessment [7]. The topic area of the maturity model is data analytics in the manufacturing environment. Thus, the model is subdivided into the primary dimension of data usage (for data analytics) and the secondary dimensions of data generation and data provision.

Currently, numerous maturity models concerning the topic areas of Industry 4.0 and digital transformation in the manufacturing environment have been published. The considered models have been developed methodically and are documented thoroughly. Relevant models are the "Industrie 4.0-Readiness-Modell" [8], the "4i-Reifegradmodell" [9] and the "I4.0 Maturity Index" [10]. These models form the basis for the derivation of the dimensions and levels of the maturity model presented in this paper. However, the above-mentioned approaches do not focus on the topic of data analytics and their dimensions in manufacturing. Therefore, these models are not ideal for the evaluation of data analytics use cases in manufacturing.

2.2 Different Types of Data Analytics

Regarding the improvement of manufacturing processes by utilizing existing data, two main approaches can be distinguished [11]. A model-driven approach requires a model of the problem to be solved that usually must be developed manually beforehand. This model embodies explicit and formalized knowledge regarding the problem. An example for a model-driven approach is operations research [11]. In operations research, the problem has to be translated into a mathematical model. It is then possible to derive a solution for the real-world problem based on this mathematical model [12]. The usage of simulation or expert systems is also model-driven [11]. Expert systems are aiming to perform similar to human capabilities

of problem solving using a database of explicit knowledge [13]. The data-driven approach of data analytics aims at the automatic generation of explicit knowledge [11]. The two approaches have different objectives, but a clear distinction can be difficult to achieve.

Within data analytics, different types can be distinguished. Descriptive analytics is already widely used and enables companies to assess their past and possibly their current business performance by utilizing historic data [14, 15]. Exploratory data analytics helps users to explore and understand data [16, 17]. Predictive analytics aims to identify patterns within historic data and extrapolate these patterns to predict further developments [15]. However, the methods of data analytics can also be used to derive historic values that are otherwise unknown [16]. Prescriptive data analytics addresses the question of the right decision which has to be made to address the predicted developments [15, 17]. Relevant tools for prescriptive data analytics are data mining [11] and model-driven approaches such as operations research, simulations and expert systems [14, 15, 17]. Moreover, some authors suggest a combination of these different approaches [16, 18]. For example, the knowledge derived from data mining methods can be used to refine an operations research model and thus improve the result [12]. Another possibility is to use the generated knowledge as input for model-driven approaches. An expert system could be used to derive a decision based on a prognosis gained from the usage of predictive data analytics.

For the development of the maturity model, the distinctions between different types of data analytics as well as the distinctions between data, information, knowledge [1], decisions and actions [5] can be utilized. Level 1 describes the usage of rudimentary descriptive data analytics. This includes standardized reports in operational information systems. The next level is descriptive and exploratory data analytics. Methods in descriptive and exploratory statistics can be used. The tools for this level also include online analytical processing (OLAP) [11, 15, 16]. OLAP enables users to interact with and navigate through data that is relevant for a decision [19]. The most relevant tool for predictive analytics is data mining [11, 15]. The application classes of data mining include cluster analysis, association rule learning, statistical classification and numerical prediction [4]. Prescriptive analytics corresponds to the term "decision" and can be matched to level 4. If decisions are implemented directly and automatically in the manufacturing environment, the final maturity level is reached [2, 9]. Figure 1 depicts the different stages and the associated tools.

2.3 Use of the Model in Manufacturing

The first and last level of the model (Fig. 2) represent the lowest and highest possible maturity level of data analytics in manufacturing [7]. Therefore, the first stage describes use cases in which no data is generated and thus the usage of data analytics is impossible. The highest maturity level describes a manufacturing environment that is self-organizing and self-optimizing [2, 9].

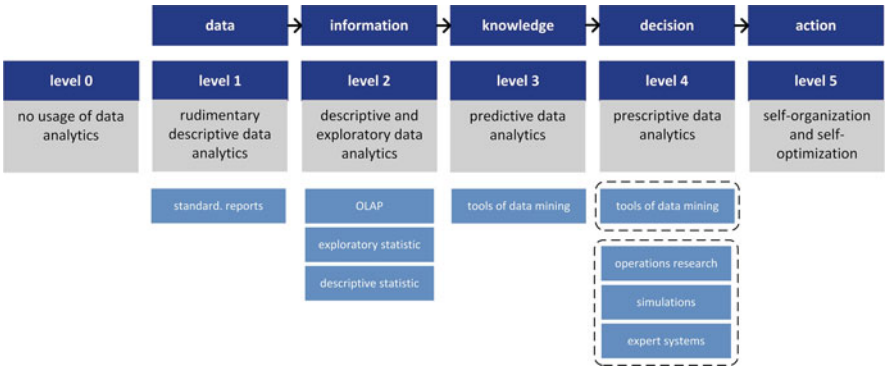


Fig. 1 Step logic and tools of data-driven and model-driven approaches (based on [1, 5])

Obviously, the data used for analytics first need to be generated. A distinction can be made between machine and process data [8]. However, the existence of data does not imply that these are available without restriction for analytical purposes. On the contrary, the access to relevant data for data analytics is still a great challenge for companies. This challenge is addressed by companies with different approaches such as data warehousing. However, this model is independent of specific approaches. Because of that, three general criteria have been defined to assess the maturity in this dimension. Approaches such as data warehousing address the issue that relevant data is scattered across different information systems. Moreover, conventional operational information systems are limited in regard to the usage of elaborate data analytics [19]. Because of that, the first criterion is the comprehensiveness of the data that is available for analytics without notable technical restriction. The next criterion describes the data quality which affects the quality of the derived knowledge significantly [4]. Furthermore, the factor time plays an important role in data analytics. One reason for this is that companies have to make vital decisions increasingly fast while the complexity of decision-making increases simultaneously [20].

The levels 4 and 5 of the dimension data analytics describe a long-term vision for the future. Because of that, the levels of the other two dimensions are designed so that the level 3 already corresponds to a high degree of fulfillment. These dimensions can be seen as prerequisites for the usage of data analytics.

3 Smart Services Use Case Example

The model is applied to a use case at TRUMPF GmbH + Co. KG - a manufacturer of machine tools and laser technology. Condition Monitoring is a service offered by TRUMPF to monitor the laser systems of customers using both algorithms and the human expertise of service engineers. Descriptive, exploratory and predictive types

	elements	level 0	level 1	level 2	level 3	level 4	level 5
data usage	data usage	no usage of data analytics	rudimentary descriptive data analytics	descriptive and exploratory data analytics	predictive data analytics	prescriptive data analytics	self-organization and self-optimization
data provision	comprehen-siveness	-	very small extent	medium extent	substantial	almost completely	No restrictions due to missing data (-sources)
	time	-	very long delay	medium delay	low delay	very low delay	no restrictions due to delays
	data quality	-	very low data quality	acceptable data quality	high data quality	very high data quality	no restrictions due to data quality
data generation	machine data	machines are not connected	individual machines connected	partially connected	predominantly connected	almost all machines are connected	all machines are connected
	process data	data are not collected	first relevant data are collected	relevant data are partially collected	relevant data are predominantly collected	almost all relevant data are collected	all relevant data are collected

Fig. 2 Maturity model for self-assessment of data analytics applications

of data analytics are being utilized to identify necessary maintenance measures. The aim is to alert customers early to avoid expensive machine failures and thus loss of production.

With regard to Condition Monitoring, only the machine data of the customer is relevant. The number of machines that are connected to the service of TRUMPF varies among the different customers. Because almost all data relevant for analytics is available, the criterion of comprehensiveness can be rated with four. Using another TRUMPF product, Factory Gate, data is transmitted in a definable time interval. The interval is set to a certain time period, so that the time delay does not limit the usage of data analytics. The data quality rarely causes problems and, therefore, can be described as very good. Automated evaluations can be executed within the Condition Monitoring portal. Different types of diagrams of descriptive statistics are used to visualize historic data over a time. However, in many cases the review of these measured values does not enable it to draw conclusions regarding the causes of error. Exploratory methods such as correlations of different measured values can be used in such cases to narrow down the root cause of failure. An even higher customer benefit is achieved by utilizing predictive analytics. The deviation from specified limit values can be predicted using linear regressions and trend analysis. In this way the failure of a component can be predicted.

Currently, the evaluation of the situation and the deduction of decisions and counteractions are not realized automatically. Instead, human knowledge that may not be available in an explicit form and direct communication with the customer is required.

There are already ideas regarding prescriptive data analytics. The service engineers document the kind of failures they solved and how they did it. This database could be deployed for decision support by using text mining. By combining predictions about possible machine outages with this knowledge, it could be possible to automatically make decisions about maintenance measures.

By using the presented model, companies can identify potentials of data analytics in their manufacturing environment. In addition to the Smart Services application example shown here, the model can also be used in other production-related applications such as production planning, logistics or spare parts management. This can be the first step to deal with the topic of data analytics, to point out possibilities to find new applications and the starting point for further developments in this area.

References

1. Rehäuser, J., Krcmar, H.: Wissensmanagement in Unternehmen. In: Schreyögg, G., Conrad, P. (eds.) Wissensmanagement, pp. 1–40. Walter de Gruyter, Berlin (1996)
2. Wissenschaftliche Gesellschaft für Produktionstechnik WGP e. V. (eds.): WGP-Standpunkt Industrie 4.0. N. p., Darmstadt (2016)
3. Gimpel, H., et al.: Structuring digital transformation – a framework of action field and its application at ZEISS. JITTA. **19**(1), 31–54 (2018)
4. Cleve, J., Lämmel, U.: Data Mining, 2nd edn. Walter de Gruyter, Berlin/Boston (2016)

5. Voß, S., Gutenschwager, K.: Informationsmanagement. Springer, Berlin/Heidelberg (2001)
6. Fraser, P., Moultrie, J., Gregory, M.: The use of maturity models/grids as a tool in assessing product development capability: a review. In: Engineering Management Conference, Cambridge/UK, 18–20 August (2002)
7. Maier, A., Moultrie, J., Clarkson, J.: Assessing organizational capabilities: reviewing and guiding the development of maturity grids. IEEE Trans. Eng. Manag. **59**(1), 138–159 (2009)
8. Lichtblau, K., et al.: Industrie 4.0-Readiness. N. p., Aachen/Köln (2015)
9. Reuter, C., et al.: Industrie 4.0 Audit. http://www.vdi-z.de/2016/Ausgabe-06/Forschung-und-Praxis/Industrie-4.0-Audit. Accessed 06 May 2018
10. Schuh, G., et al. (eds.): Industrie 4.0 Maturity Index: Die digitale Transformation von Unternehmen gestalten (acatech STUDIE). Hubert Utz Verlag, München (2017)
11. Gröger, C.: Advanced Manufacturing Analytics. Datengetriebene Optimierung von Fertigungsprozessen. Dissertation, Universität Stuttgart, Josef Eul Verlag, Lohmar (2015)
12. Meisel, S., Mattfeld, D.: Synergies of operations research and data mining. Eur. J. Oper. Res. **206**(1), 1–10 (2010)
13. Kurbel, K.: Entwicklung und Einsatz von Expertensystemen. Eine anwendungsorientierte Einführung in wissensbasierte Systeme, 2nd edn. Springer-Verlag, Berlin/Heidelberg (1992)
14. Delen, D., Demirkan, H.: Data, information and analytics as services. Decis. Support. Syst. **55**(1), 359–363 (2013)
15. Lustig, I., et al.: The analytics journey - an IBM view of the structured data analysis landscape: descriptive, predictive and prescriptive analytics. http://analytics-magazine.org/the-analytics-journey/. Accessed 06 May 2018
16. Lanquillon, C., Mallow, H.: Advanced analytics mit big data. In: Dorschel, J. (ed.) Praxishandbuch Big Data. Wirtschaft – Recht – Technik, pp. 55–89. Springer Gabler, Wiesbaden (2015)
17. Freitag, M., et al.: Potenziale von Data Science in Produktion und Logistik. Teil 1 – Eine Einführung in aktuelle Ansätze der Data Science. Industrie 4.0. Management. **31**(5), 22–26 (2015)
18. Evans, J.: Business analytics: the next frontier for decision sciences. Decision Line. **43**(2), 4–6 (2012)
19. Hannig, U.: Knowledge management + business intelligence = decision intelligence. In: Hannig, U. (ed.) Knowledge Management und Business Intelligence, pp. 3–25. Springer, Berlin/Heidelberg (2002)
20. Bleicher, K.: Das Konzept Integriertes Management. Visionen – Missionen – Programme. Campus Verlag, Frankfurt (2011)

Part III
Decision Theory and Multiple Criteria Decision Making

Fair Resource Allocation by Gini Index Minimization

Wlodzimierz Ogryczak and Grzegorz Zalewski

1 Introduction

Resource allocation problems are concerned with the allocation of limited resources among competing activities so as to achieve the best performances of all activities. The generic resource allocation problem may be stated as follows. There is a system dealing with a set I of m entities (services, activities, agents). There is given a set Q of allocation patterns (allocation decisions). For each entity $i \in I$ a function $f_i(\mathbf{x})$ of the allocation pattern $\mathbf{x} \in Q$ is defined, which measures the outcome (effect) $y_i = f_i(\mathbf{x})$ of allocation pattern $\mathbf{x}$ for entity i. In most applications, a larger value of the outcome usually means a better effect (higher service quality). Otherwise, the outcomes can be replaced with their complements to some large number. Thus, we get a vector maximization problem:

$$\max \{\mathbf{f}(\mathbf{x}) \ : \ \mathbf{x} \in Q\} \tag{1}$$

where $\mathbf{f}(\mathbf{x})$ is a vector-function that maps the decision space $X = R^n$ into the criterion space $Y = R^m$, and $Q \subset X$ denotes the feasible set.

The mean (or sum) aggregation of individual outcomes

$$\mu(\mathbf{y}) = \frac{1}{m} \sum_{i \in I} y_i \tag{2}$$

W. Ogryczak (✉)
Warsaw University of Technology, Warsaw, Poland
e-mail: W.Ogryczak@elka.pw.edu.pl; wogrycza@elka.pw.edu.pl;
http://www.ia.pw.edu.pl/~wogrycza/

G. Zalewski
National Institute of Telecommunications, Warsaw, Poland

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_11

is commonly used to search for efficient allocation patterns by solving the scalar optimization:

$$\max\{\mu(\mathbf{f}(\mathbf{x})) \ : \ \mathbf{x} \in Q\} \tag{3}$$

However, in systems which serve many users there is a need to respect some fairness (equity) rules [8, 11] while looking for the overall efficiency. Problems of fair allocation decisions arise also in technical systems like in telecommunication networks which must serve various users/demands [18, 19]. The concepts of fair optimization based on the multiple criteria equitable optimization can effectively be used to generate various fair and efficient allocation schemes. The concept of fair optimization is a specific refinement of the Pareto-optimality taking into account the inequality minimization according to the Pigou-Dalton approach [9, 12]. First of all, the fairness requires impartiality of evaluation, thus focusing on the distribution of outcome values while ignoring their ordering. Hence, the preference model is impartial (anonymous, symmetric), i.e.,

$$(y_{\pi(1)}, \ldots, y_{\pi(m)}) \cong (y_1, \ldots, y_m) \quad \forall\, \pi \in \Pi(I) \tag{4}$$

where $\Pi(I)$ denotes the set of all permutations of I. This means that any permuted outcome vector is indifferent in terms of the preference relation. Further, fairness requires equitability of outcomes which causes that the preference model should satisfy the (Pigou–Dalton) principle of transfers. The principle of transfers states that a transfer of any small amount from an outcome to any other relatively worse-off outcome results in a more preferred outcome vector, i.e., whenever $y_{i'} > y_{i''}$ then

$$\mathbf{y} - \varepsilon\mathbf{e}_{i'} + \varepsilon\mathbf{e}_{i''} \succ \mathbf{y} \quad \text{for } 0 < \varepsilon < (y_{i'} - y_{i''}) \tag{5}$$

The rational preference relations satisfying additionally properties (4) and (5) are called *fair (equitable) rational*. Outcome vector $\mathbf{y}'$ *fairly dominates* $\mathbf{y}''$, ($\mathbf{y}' \succ_e \mathbf{y}''$), iff $\mathbf{y}'$ is preferred to $\mathbf{y}''$ for all fair rational preference relations. In other words, $\mathbf{y}'$ fairly dominates $\mathbf{y}''$, if there exists a finite sequence of vectors $\mathbf{y}^j$ ($j = 1, 2, \ldots, s$) such that $\mathbf{y}^1 = \mathbf{y}''$, $\mathbf{y}^s = \mathbf{y}'$ and $\mathbf{y}^j$ is constructed from $\mathbf{y}^{j-1}$ by application of either permutation of coordinates, equitable transfer, or increase of a coordinate. An allocation pattern $\mathbf{x} \in Q$ is called *fairly optimal* or *fairly efficient* if $\mathbf{y} = \mathbf{f}(\mathbf{x})$ is fairly nondominated. Every fairly optimal solution is also Pareto-optimal, but not vice verse. Fair optimization depends on finding fairly optimal solutions.

In system analysis fairness is usually quantified with so-called fairness measures (or inequality measures), which are functions ϱ that maps $\mathbf{y}$ into (nonnegative) real numbers. Various measures have been proposed throughout the years, e.g., in [1, 3, 7, 10, 14, 16, 17] and references therein. Typical inequality measures are deviation type dispersion characteristics. They are *translation invariant* in the sense that $\varrho(\mathbf{y} + a\mathbf{e}) = \varrho(\mathbf{y})$ for any real number a (where $\mathbf{e}$ vector of units $(1, \ldots, 1)$), thus being not affected by any shift of the outcome scale. Moreover, the inequality measures are

also *inequality relevant* which means that they are equal to 0 in the case of perfectly equal outcomes while taking positive values for unequal ones, thus to be minimized for fairness.

The simplest inequality measures are based on the absolute measurement of the spread of outcomes, like the *mean absolute difference*

$$\Gamma(\mathbf{y}) = \frac{1}{2m^2}\sum_{i\in I}\sum_{j\in I}|y_i - y_j| = \mu(\mathbf{y}) - \frac{1}{m^2}\sum_{i\in I}\sum_{j\in I}\min\{y_i, y_j\}. \tag{6}$$

Another group of measures is related to deviations from the mean outcome, like the *mean absolute deviation*

$$\delta(\mathbf{y}) = \frac{1}{m}\sum_{i\in I}|y_i - \mu(\mathbf{y})|. \tag{7}$$

The *standard deviation* σ (or the *variance* σ^2) represents both the deviations and the spread measurement as

$$\sigma^2(\mathbf{y}) = \frac{1}{m}\sum_{i\in I}(y_i - \mu(\mathbf{y}))^2 = \frac{1}{2m^2}\sum_{i\in I}\sum_{j\in I}(y_i - y_j)^2. \tag{8}$$

In economics there are usually used relative inequality measures, i.e. ratio measures with the absolute inequality measures normalized by the mean outcome, so-called indices. The most commonly accepted is the Gini index (Gini coefficient) [6]

$$G(\mathbf{y}) = \frac{\Gamma(\mathbf{y})}{\mu(\mathbf{y})}, \tag{9}$$

which is the relative mean difference. Considered in networking the Jain's index [7] is based on the normalized standard deviation (the coefficient of variance). When applied to resource allocation models, the Gini index measures the relative mean absolute difference of outcomes [13, 20]. In order to achieve equitable solution the best allocation scheme is defined as the one that minimizes the Gini index of the outcomes. It was empirically found in real-life applications [5] that this objective while equalizing the outcomes (fairness) it may simultaneously support their maximization (efficiency). Although it depends on the feasible set structure and there is no guarantee to achieve good equitable and efficient allocation scheme as for some feasible sets the outcomes may be minimized [4] thus leading to inferior conclusions with respect to the outcomes maximization. The class of preference models complying with the optimization of outcomes as well as with an equal consideration of the activities is mathematically formalized with the concept of equitable dominance [14, 16]. Solution concepts equitably consistent do not contradict the maximization of outcomes or the inequality minimization.

Therefore, the achievement of equitable consistency by the mean-equity models has a paramount importance. In this paper we show that when combined with the reference point model, the Gini index minimization is consistent with equitable maximization preferences. When appropriate outcome shift is applied, then the Gini index minimization is consistent both with inequity minimization and with outcomes maximization thus guaranteeing equitable allocation schemes.

2 The Main Result

Let us consider the outcomes with a shift Δ (or equivalently minimum target outcome and measured deviations) $\tilde{y}_i = y_i - \Delta$ and the corresponding Gini index minimization

$$\min \frac{\Gamma(\tilde{\mathbf{y}})}{\mu(\tilde{\mathbf{y}})} = \frac{\Gamma(\mathbf{y})}{\mu(\mathbf{y}) - \Delta} \quad \text{for } \mu(\mathbf{y}) > \Delta \tag{10}$$

We show that such Gini index minimization (10) is consistent both with inequity minimization and with outcomes maximization provided that $\Delta \geq \mu(\mathbf{y}) - \Gamma(\mathbf{y}) = \frac{1}{m^2} \sum_{i \in I} \sum_{j \in I} \min\{y_i, y_j\}$ at the optimum. Thus with minor clear restrictions on target selection Δ the Gini index minimization guarantees fairly efficient solution. Exactly, we consider the regularized ratio optimization model with constraint on the mean value transformed into the weak inequality with tolerance ε as well as accordingly regularized the numerator (mean absolute difference value):

$$\min \left\{ \frac{\Gamma(\mathbf{f}(\mathbf{x})) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x})) - \Delta} \ : \ \mu(\mathbf{f}(\mathbf{x})) \geq \Delta + \varepsilon, \ \mathbf{x} \in Q \right\} \tag{11}$$

where ε is an arbitrary small positive number. The numerator regularization is important when for multiple allocation patterns the inequality measure $\Gamma(\mathbf{f}(\mathbf{x}))$ takes value equal to zero. In these cases, an optimal solution to problem (11) is the allocation with the largest mean outcome. Obviously, the value ε introduced in the numerator can be, in principle, different from the one included in the problem constraint. However, to the sake of a simple exposition, we decided to use the same notation for both. The following theorem is valid.

Theorem 1 *If* $\mathbf{x}^0$ *is an optimal solution to the ratio optimization (11) and it satisfies* $\mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0)) \leq \Delta$, *then* $\mathbf{x}^0$ *is nondominated in terms of the bi-criteria maximization* $\max\{\mu(\mathbf{f}(\mathbf{x})), \mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x}))\}$, *as well as in terms of the bi-criteria mean-equity optimization* $\max\{\mu(\mathbf{f}(\mathbf{x})), -\Gamma(\mathbf{f}(\mathbf{x}))\}$.

Proof Suppose that there exists a feasible vector $\mathbf{x}$, i.e., $\mathbf{x} \in Q$ and $\mu(\mathbf{f}(\mathbf{x})) - \Delta \geq \varepsilon$, such that $\mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x})) \geq \mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0))$ and $\mu(\mathbf{f}(\mathbf{x})) \geq \mu(\mathbf{f}(\mathbf{x}^0))$. Note that the objective function in problem (11) can be written as:

$$\frac{\Gamma(\mathbf{f}(\mathbf{x})) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x})) - \Delta} = \frac{\Delta - (\mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x}))) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x})) - \Delta} + 1.$$

Note that due to the optimality of $\mathbf{x}^0$ and the additional condition $\mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0)) \leq \Delta$, in the above ratio both numerator and denominator are positive for solution $\mathbf{x}^0$, whereas the denominator is positive for any feasible $\mathbf{x}$. Hence, whenever $\mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x})) > \mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0))$ or $\mu(\mathbf{f}(\mathbf{x})) > \mu(\mathbf{f}(\mathbf{x}^0))$, the following inequality holds:

$$\begin{aligned}\frac{\Gamma(\mathbf{f}(\mathbf{x})) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x})) - \Delta} &= \frac{\Delta - (\mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x}))) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x})) - \Delta} + 1 \\ &< \frac{\Delta - (\mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0))) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x}^0)) - \Delta} + 1 = \frac{\Gamma(\mathbf{f}(\mathbf{x}^0)) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x}^0)) - \Delta}\end{aligned}$$

which contradicts the optimality of $\mathbf{x}^0$. Therefore, $\mu(\mathbf{f}(\mathbf{x})) = \mu(\mathbf{f}(\mathbf{x}^0))$ and $\Gamma(\mathbf{f}(\mathbf{x})) = \Gamma(\mathbf{f}(\mathbf{x}^0))$, $\mathbf{x}$ is an equivalent optimal solution to (11), and solution $\mathbf{x}^0$ is nondominated in terms of the bi-criteria maximization $\max\{\mu(\mathbf{f}(\mathbf{x})), \mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x}))\}$.

Finally, suppose a feasible solution $\mathbf{x}$ dominates $\mathbf{x}^0$ in terms of bi-criteria mean-equity optimization $\max\{\mu(\mathbf{f}(\mathbf{x})), -\Gamma(\mathbf{f}(\mathbf{x}))\}$, i.e., $\Gamma(\mathbf{f}(\mathbf{x})) \leq \Gamma(\mathbf{f}(\mathbf{x}^0))$ and $\mu(\mathbf{f}(\mathbf{x})) \geq \mu(\mathbf{f}(\mathbf{x}^0))$ with at least one strict inequality. Then, the two conditions $\mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x})) \geq \mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0))$ and $\mu(\mathbf{f}(\mathbf{x})) \geq \mu(\mathbf{f}(\mathbf{x}^0))$ hold, with at least one strict inequality. Hence, optimal solution $\mathbf{x}^0$ is also nondominated in terms of bi-criteria mean-equity optimization $\max\{\mu(\mathbf{f}(\mathbf{x})), -\Gamma(\mathbf{f}(\mathbf{x}))\}$. □

Corollary 1 *If $\mathbf{x}^0$ is an optimal solution to the ratio optimization problem (11) that satisfies condition $\mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0)) \leq \Delta$, then $\mathbf{x}^0$ is a fairly nondominated solution.*

Proof Suppose that there exists a feasible solution $\mathbf{x}$, i.e., $\mathbf{x} \in Q$ and $\mu(\mathbf{f}(\mathbf{x})) \geq \Delta + \varepsilon$, such that $\mathbf{x} \succ_e \mathbf{x}^0$. The latter relation implies [15] that $\mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x})) > \mu(\mathbf{f}(\mathbf{x}^0)) - \Gamma(\mathbf{f}(\mathbf{x}^0))$ and $\mu(\mathbf{f}(\mathbf{x})) \geq \mu(\mathbf{f}(\mathbf{x}^0))$. Therefore, $\mathbf{x}^0$ is dominated by $\mathbf{x}$ in terms of the bi-criteria maximization $\max\{\mu(\mathbf{f}(\mathbf{x})), \mu(\mathbf{f}(\mathbf{x})) - \Gamma(\mathbf{f}(\mathbf{x}))\}$, which contradicts Theorem 1. Hence, $\mathbf{x}^0$ is a fairly nondominated solution. □

The interval of appropriate target values depends on the allocation problem structure (feasible set). Although it can be find or adjusted during the optimization process without necessity of a special feasible set analysis. Moreover, the usage of the target value allows one to model various fairness preferences. The ratio optimization problem (11) can easily be linearized in the case of linear constraints [2] as well as mixed integer ones [21].

Note that Theorem 1 can easily be generalized for some other deviation type inequality measures ϱ. In particular, for the mean absolute deviation (7). Hence, for the relative mean absolute deviation $S(\mathbf{y}) = \delta(\mathbf{y})/(2\mu(\mathbf{y}))$, called Schutz Index (known also as Hoover Index, Robin Hood Index, Pietra Ratio) can be shown similar fair efficiency results as for the Gini index.

Corollary 2 *If* $\mathbf{x}^0$ *is an optimal solution to the ratio optimization problem*

$$\min \left\{ \frac{0.5\delta(\mathbf{f}(\mathbf{x})) + \varepsilon}{\mu(\mathbf{f}(\mathbf{x})) - \Delta} \ : \ \mu(\mathbf{f}(\mathbf{x})) \geq \Delta + \varepsilon, \ \mathbf{x} \in Q \right\} \tag{12}$$

that satisfies condition $\mu(\mathbf{f}(\mathbf{x}^0)) - 0.5\delta(\mathbf{f}(\mathbf{x}^0)) \leq \Delta$*, then* $\mathbf{x}^0$ *is fairly nondominated with the exception of alternative (and equivalent) optimal solutions having the same values of mean outcome* $\mu(\mathbf{f}(\mathbf{x}^0))$ *and mean absolute deviation* $\bar{\delta}(\mathbf{f}(\mathbf{x}^0))$.

Unfortunately, such results are not valid for the coefficient of variance, and therefore for the Jain's index optimization.

References

1. Atkinson, A.B.: On the measurement of inequality. J. Econ. Theory **2**, 244–263 (1970)
2. Charnes, A., Cooper, W.: Programming with linear fractional functionals. Naval Res. Logist. Q. **9**(3–4), 181–186 (1962)
3. Dianati, M., Shen, X., Naik, S.: A new fairness index for radio resource allocation in wireless networks. In: IEEE WCNC 2005, vol. 2, pp. 712–717
4. Drezner, T., Drezner, Z., Guyse, J.: Equitable service by a facility: minimizing the Gini coefficient. Comput. OR **36**, 3240–3246 (2009)
5. Erkut, E.: Inequality measures for location problems. Locat. Sci. **1**, 199–217 (1993)
6. Gastwirth, J.L.: The estimation of the Lorenz curve and Gini index. Rev. Econ. Stat. **54**, 306–316 (1972)
7. Jain, R., Chiu, D., Hawe, W.: A Quantitative Measure of Fairness and Discrimination for Resource Allocation in Shared Computer System. ERL, Digital Equipment, Hudson (1984)
8. Karsu, Ö., Morton, A.: Inequity averse optimisation in operational research. Eur. J. Oper. Res. **245**(2), 343–359 (2015)
9. Kostreva, M.M., Ogryczak, W., Wierzbicki, A.: Equitable aggregations and multiple criteria analysis. Eur. J. Oper. Res. **158**, 362–367 (2004)
10. Lan, T., Kao, D., Chiang, M., Sabharwal, A.: An axiomatic theory of fairness in network resource allocation. In: IEEE INFOCOM 2010, pp. 1–9.
11. Luss, H.: Equitable Resource Allocation: Models, Algorithms, and Applications. Wiley, Hoboken (2012)
12. Marshall, A.W., Olkin, I.: Inequalities: Theory of Majorization and Its Applications. Academic, New York (1979)
13. Mulligan, G.F.: Equity measures and facility location. Pap. Reg. Sci. **70**, 345–365 (1991)
14. Ogryczak, W.: Inequality measures and equitable approaches to location problems. Eur. J. Oper. Res. **122**, 374–391 (2000)
15. Ogryczak, W.: Multicriteria models for fair resource allocation. Control. Cybern. **36**, 303–332 (2007)
16. Ogryczak, W.: Inequality measures and equitable locations. Ann. Oper. Res. **167**, 61–86 (2009)

17. Ogryczak, W.: Fair optimization - methodological foundations of fairness in network resource allocation. In: IEEE 38th COMPSACW, pp. 43–48, Vasteras (2014)
18. Ogryczak, W., Luss, H., Pióro, M., Nace, D., Tomaszewski, A.: Fair optimization and networks: a survey. J. Appl. Math. **2014**. Article ID 612018, 25 (2014)
19. Pióro, M., Medhi, D.: Routing, Flow and Capacity Design in Communication and Computer Networks. Morgan Kaufmann, San Francisco (2004)
20. Tao, Y., Henry, K., Zou, Q., Zhong, X.: Methods for measuring horizontal equity in health resource allocation: a comparative study. Health Econ. Rev. **4**, 1–10 (2014)
21. Williams, H.P.: Experiments in the formulation of integer programming problems. Math. Program. Study **2**, 180–197 (1974)

Part IV
Discrete and Integer Optimization

A Sweep-Plane Algorithm for the Computation of the Volume of a Union of Polytopes

Lovis Anderson and Benjamin Hiller

1 Introduction

A *(polyhedral) disjunctive set* is a set that may be written as a union of polytopes $\mathcal{P} = \bigcup_{i=1}^{r} P^i$. Disjunctive sets are a frequently occurring structure in many optimization models. As they are in general nonconvex sets, global optimization solvers rely on convex relaxations to obtain bounds. Using tight convex relaxations during the solution process is crucial to be able to solve such models. The tightness of the convex relaxation, i.e. the convexification error, may be naturally measured using the volume. The volume may be used to compare two competing convex relaxations [5], or to numerically quantify the convexification error by comparing the volume of $\mathcal{P}$ and its convex relaxation. This may be used to devise a branching scheme for $\mathcal{P}$ that quickly improves the tightness of the relaxation [3].

Computing the convexification error numerically requires computing the volume of the disjunctive set $\mathcal{P}$. Naively, this can be done via inclusion/exclusion and leveraging the well-studied algorithms for computing the volume of a single polytope. However, this is often inefficient. Bieri and Nef [1] proposed a method for computing the volume of a disjunctive set which calculates the volume up to a sweep-plane in a *beneath and beyond* approach. The sweep-plane is a hyperplane given by $\{\boldsymbol{x} \mid \langle \boldsymbol{s}, \boldsymbol{x} \rangle = \lambda\}$ where $\boldsymbol{s}$ is the sweep-direction and λ is the time in the sweep. Similar to a tomography, the sweep-plane is moved through space, as visualized by Fig. 1. The sweep-plane volume function $f(\lambda)$ describes the volume $\mathrm{Vol}(\mathcal{P} \cap \{\boldsymbol{x} \mid \langle \boldsymbol{s}, \boldsymbol{x} \rangle \leq \lambda\})$ that lies behind the sweep-plane. This function allows us to see interesting changes in the volume over the course of the sweep, which may be useful to guide decomposition methods as in [3].

L. Anderson (✉) · B. Hiller
Zuse Institute, Berlin, Germany
e-mail: anderson@zib.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_12

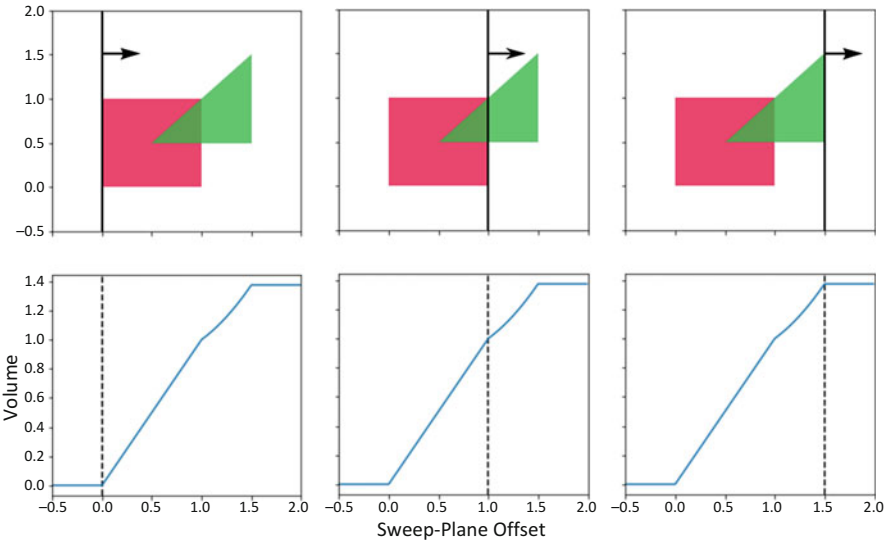


Fig. 1 Top row: a union of polytopes with sweep-planes at different times. Bottom row: the corresponding sweep-plane volume graphs

We propose a revised algorithm based on Bieri and Nefs general idea. First we review a formula for the volume of a polytope as introduced in [4]. Afterwards we extend the formula to unions of polytopes and discuss the resulting algorithm. A preliminary python implementation of the proposed algorithm can be found at https://gitlab.com/LovisAnderson/sweepvolume.

2 The Volume of a Polytope

We will first introduce a formula for the volume of a single polytope P, which is given by $P := \{\boldsymbol{x} \in \mathbb{R}^d \mid Ax \leq b\}$. We will restrict ourselves to simple polytopes, i.e. polytopes for which at each vertex exactly dimensional many constraints are active. The non-simple case can either be handled by perturbation of the constraints or regularization as described in [1].

2.1 The Conic Decomposition

The *vertex cone* $K_{\boldsymbol{v}}$ at a vertex $\boldsymbol{v}$ is given by the constraints that are active at that vertex. We can also describe that cone as a Minkowski sum of the vertex and the positive hull of its rays. In particular the rays on the d edges suffice to describe the

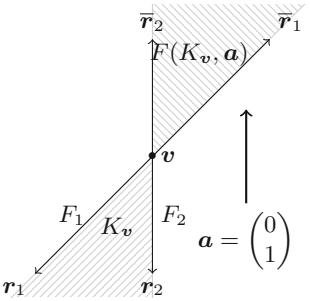


Fig. 2 The vertex cone and the forward cone

positive hull. We can write $K_v = \{v\} + \text{pos}(r_1, \ldots, r_d)$. The rays $r_1, \ldots, r_d$ are unique up to scaling and can be calculated from the constraint matrix. Let A_v be the matrix that is given by the constraints that are active at v. Then the rays are given by

$$(r_1, \ldots, r_d) = -A_v^{-1}. \tag{1}$$

A *forward cone* $F(K_v, s)$ for a vertex cone s results from flipping the rays such that they point in the same direction as s. Formally,

$$F(K_v, s) := \{v\} + \text{pos}(\overline{r}_1), \ldots, \overline{r}_d) \quad \text{with} \quad \overline{r}_i := \begin{cases} r_i & \text{if } \langle r_i, s \rangle > 0, \\ -r_i & \text{else} \end{cases}$$

Figure 2 visualizes the vertex cone and forward cone in an exemplary way.

Theorem 1 is a slight abbreviation of the main result from [4] and shows how a polytope can be decomposed into its forward cones using the indicator function $\mathbb{1}$.

Theorem 1 *Let P be a simple polytope and s be a direction which induces a total ordering $v_1, \ldots, v_n$ on the vertices of P via the dot product, i.e. $\langle v_i, s \rangle < \langle v_j, s \rangle$ for $i < j$. Let K_i be the vertex cone at v_i and $\sigma(K_i, s) := \#\{i \in \{1, \ldots, d\} \mid \langle r_i, s \rangle < 0\}$ be the number of rays that have to be flipped in order to obtain the forward cone $F(K_i, s)$ from K_i. Then we have*

$$\mathbb{1}_P = \sum_{i \in [n]} (-1)^{\sigma(K_i, s)} \mathbb{1}_{F(K_i, s)}. \tag{2}$$

We are interested in the volume up to a sweep-plane, which depends on the direction s and a *time* (or geometrically interpreted *offset*) λ. We denote the *negative halfspace* at time λ by $H^-(s, \lambda) := \{x \in \mathbb{R}^d \mid \langle s, x \rangle \leq \lambda\}$. Through integration we can use Theorem 1 to write the volume up to the sweep-plane at time λ as

$$\text{Vol}(P \cap H^-(s, \lambda)) = \sum_{i \in [n]} (-1)^{\sigma(K_i, s)} \text{Vol}(F(K_i, s) \cap H^-(s, \lambda)). \tag{3}$$

2.2 The Sweep-Plane Volume of a Cone

To be able to compute the volume of a polytope by using Eq. (3) we need a way to compute the sweep-plane volume of a forward cone. The volume of a simplex might be calculated with this neat determinant formula, see [6].

Fact 1 *The volume of a simplex $S \subset \mathbb{R}^d$ with vertices $\boldsymbol{s}_1, \ldots, \boldsymbol{s}_{d+1}$ is:*

$$\mathrm{Vol}(S) = \frac{1}{d!} |\det\left(\left(\boldsymbol{s}_2 - \boldsymbol{s}_1 \ \ldots \ \boldsymbol{s}_{d+1} - \boldsymbol{s}_1\right)\right)| \tag{4}$$

We can adopt Eq. (4) such that we can use it to compute the sweep-plane volume of a forward cone. The forward cone up to a sweep-plane is a simplex which can be described by its vertex and rays. In this simplex, the offset λ of the sweep is only reflected in the length of the rays and not in their direction.

Proposition 1 *Let $K_{\boldsymbol{v}} = \{\boldsymbol{v}\} + \mathrm{pos}(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)$ be a vertex cone of a polytope P. Let $\boldsymbol{s}$ be such that it induces a total ordering on the vertices of P. Let $\lambda_{\boldsymbol{v}} := \langle \boldsymbol{s}, \boldsymbol{v} \rangle$. Then for the volume of the forward cone $F(K_{\boldsymbol{v}}, \boldsymbol{s})$ and $\lambda > \lambda_{\boldsymbol{v}}$ it holds that*

$$\mathrm{Vol}(F(K_{\boldsymbol{v}}, \boldsymbol{s}) \cap H^-(\boldsymbol{s}, \lambda)) = (-1)^{\sigma(K_{\boldsymbol{v}}, \boldsymbol{s})} (\lambda - \lambda_{\boldsymbol{v}})^d \frac{|\det(\boldsymbol{r}_1, \ldots, \boldsymbol{r}_d)|}{d! \prod_{i \in [d]} \langle \boldsymbol{s}, \boldsymbol{r}_i \rangle}. \tag{5}$$

For $\lambda \leq \lambda_{\boldsymbol{v}}$ clearly holds $\mathrm{Vol}(F(K_{\boldsymbol{v}}, \boldsymbol{s}) \cap H^-(\boldsymbol{s}, \lambda)) = 0$.

We need the vertices of P to be strictly ordered with respect to $\boldsymbol{s}$ to assure that the dot product in Proposition 1 is not 0. However, this can always be achieved by perturbing $\boldsymbol{s}$.

3 Extension to Unions of Polytopes

We want to extend our decomposition to a union of polytopes $\mathcal{P} = \bigcup_{i=1}^{r} P^i$. We now look at all constraints which describe polytopes in $\mathcal{P}$ and assume we have them ordered as $\{\boldsymbol{a}_i^T \boldsymbol{x} \leq b_i, i \in \{1, \ldots, n\}\}$. We call $\mathcal{A} := \{(\boldsymbol{a}_i, b_i) \mid i \in \{1, \ldots, n\}\}$ a *hyperplane arrangement*. The corresponding equality constraints $\{\boldsymbol{a}_i^T \boldsymbol{x} = b_i \mid i \in \{1, \ldots, n\}\}$ decompose the space into *cells*, which are defined as the closures of the connected components of

$$\mathbb{R}^d \setminus \{\boldsymbol{x} \in \mathbb{R}^d \mid \exists i \text{s.t.} \boldsymbol{a}_i^T \boldsymbol{x} = b_i\}.$$

That means that the cells are the (closed) sets which are not intersected by a hyperplane which is given by an equality constraint, see Fig. 3. The union of polytopes can be written as the union of a subset of these cells. All cells are polytopes and are mutually internally disjoint. Therefore the sum of the volumes

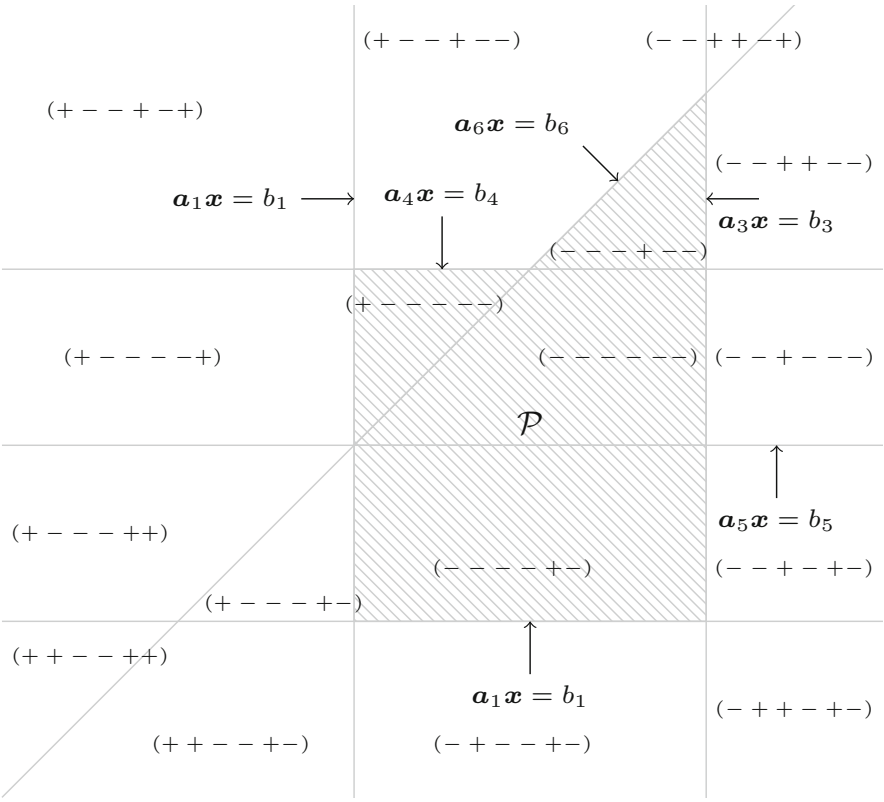


Fig. 3 The hyperplane arrangement $\mathcal{A}(\mathcal{P})$, with $\mathcal{P}$ being the union of a square and a triangle and the region describing position vectors

of the cells which are subsets of the union of polytopes equals the total volume of the union of polytopes.

Proposition 2 *Let* $\mathcal{P} = \bigcup_{i=1}^{r} P_i$ *be a union of polytopes and* $\mathcal{A}_\mathcal{P}$ *be the corresponding hyperplane arrangement. Let* $C_1, \ldots, C_l$ *be the cells of* $\mathcal{A}_\mathcal{P}$ *which are subsets of* $\mathcal{P}$*. We can then write* $\mathrm{Vol}(\mathcal{P}) = \sum_{i=1}^{l} \mathrm{Vol}(C_i)$.

The following definition generalizes the well-known concept of a position vector [2] and helps us to describe cells and other faces of a hyperplane arrangement.

Definition 1 Let $\mathcal{A} := \{(\boldsymbol{a}_i, b_i) \mid i \in \{1, \ldots, m\}\}$ be a hyperplane arrangement. We call a vector $(p_1, \ldots, p_m) =: \boldsymbol{p} \in \{+, -, 0, \bullet\}^m$ a *position vector* for $\mathcal{A}$. We associate with it the set

$$\mathcal{A}(\boldsymbol{p}) := \{\boldsymbol{x} \in \mathbb{R}^d \mid \boldsymbol{a}_i^T \boldsymbol{x} \begin{cases} \leq b_i & \text{if} p_i = -, \\ \geq b_i & \text{if} p_i = +, \\ = b_i & \text{if} p_i = 0, \end{cases} \quad i \in \{1, \ldots, m\}\}.$$

To ensure that no two position vectors correspond to the same set we also demand that for all $i \in [m]$ with $p_i = \cdot$ there is a $\boldsymbol{x}_1 \in \mathcal{A}(\boldsymbol{p})$ with $\boldsymbol{a}_i^T \boldsymbol{x}_1 < b_i$ and a $\boldsymbol{x}_2 \in \mathcal{A}(\boldsymbol{p})$ with $\boldsymbol{a}_i^T \boldsymbol{x}_2 > b_i$. If there is no such pair $\boldsymbol{x}_1, \boldsymbol{x}_2$ we can replace $\cdot$ with $+$ or $-$.

We can describe any face of the hyperplane arrangement using position vectors. Including $\cdot$ in the definition also allows us to describe polytopes and vertex cones. This allows us to compare those sets in a combinatorial way. For position vectors $\boldsymbol{p}^1, \boldsymbol{p}^2$ we say that $\boldsymbol{p}^1$ is *included* in $\boldsymbol{p}^2$ if

$$\begin{aligned} p_i^1 = p_i^2 \vee p_i^2 = \cdot \qquad & \text{if} \quad p_i^1 \in \{+, -\} \\ p_i^2 = \cdot \qquad & \text{if} \quad p_i^1 = \cdot. \end{aligned}$$

If $\boldsymbol{p}^1$ is included in $(\boldsymbol{p}^2)$ then $\mathcal{A}(\boldsymbol{p}^1) \subseteq \mathcal{A}(\boldsymbol{p}^2)$ and we write $\boldsymbol{p}^1 \subseteq \boldsymbol{p}^2$.

Corollary 1 *Let $\mathcal{P} = \bigcup_{i=1}^r$ be a union of polytopes and $\mathcal{A}_\mathcal{P}$ be the corresponding hyperplane arrangement. Let $C_1, \ldots, C_l$ be the cells of $\mathcal{A}_\mathcal{P}$ which are subsets of $\mathcal{P}$. Let $\boldsymbol{v}$ be a vertex whose active constraints in the arrangement are given by the indices $J := \{j_1, \ldots, j_d\}$. We define the restricted position vector of $K_{\boldsymbol{v}}$ only featuring the active constraints at $\boldsymbol{v}$ as $\boldsymbol{p}^{|\boldsymbol{v}}(K_{\boldsymbol{v}}) := (p_1^{|\boldsymbol{v}}, \ldots, p_m^{|\boldsymbol{v}})$ with $p_i^{|\boldsymbol{v}} := \begin{cases} p_i(K_{\boldsymbol{v}}) & \text{if} \quad i \in J \\ \cdot & \text{else} \end{cases}$. A cone $K_{\boldsymbol{v}}$ at a vertex $\boldsymbol{v}$ of the hyperplane arrangement is the vertex cone of some cell C_j if and only if there is a P_i such that*

$$\boldsymbol{p}(\boldsymbol{v}) \subseteq \boldsymbol{p}(P_i) \qquad \wedge \qquad \boldsymbol{p}^{|\boldsymbol{v}}(K_{\boldsymbol{v}}) \subseteq \boldsymbol{p}(P_i). \tag{6}$$

This allows us to algorithmically find all vertex cones of cells belonging to $\mathcal{P}$.

4 The Algorithm

The steps in the algorithm for computing the sweep-plane volume function for a union of polytopes $\mathcal{P} := \bigcup_{i=1}^r P_i$ can be summarized as follows:

1. Calculate the vertices of the hyperplane arrangement
2. For each vertex calculate the vertex cones belonging to $\mathcal{P}$ using Corollary 1
3. Calculate the sweep-plane volume function using Proposition 1

In our implementation we do steps 1 and 2 independently of step 3. We build a data structure which we can then use in step 3 to calculate the sweep-plane volume function. Almost all of the computational complexity lies in building the data structure. Once the data structure is built, we are able to calculate sweep-plane volume functions for many different directions. Through comparing these functions, insights into the structure of the disjunctive set might be obtained. Detecting

structural changes in volume might be useful to guide decomposition methods as in [3].

Acknowledgements The authors thank the DFG for support within project A04 in CRC TRR154 and the BMBF Research Campus Modal (fund number 05M14ZAM) and ICT COST Action TD1207 for additional support.

References

1. Bieri, H., Nef, W.: A sweep-plane algorithm for computing the volume of polyhedra represented in boolean form. Linear Algebra Appl. **52**, 69–97 (1983)
2. Edelsbrunner, H.: Algorithms in Combinatorial Geometry. Springer, Berlin (1987)
3. Hiller, B., Walther, T.: Improving branching for disjunctive polyhedral models using approximate convex decompositions, ZIB-Report 17-68, Zuse Institute Berlin, 2017
4. Lawrence, J.: Polytope volume computation. Math. Comput. **57**(195), 259–271 (1991)
5. Lee, J., Skipper, D., Speakman, E.: Algorithmic and modeling insights via volumetric comparison of polyhedral relaxations. Math. Program. Ser. B **170**(1), 121–140 (2018)
6. Stein, P.: A note on the volume of a simplex. Am. Math. Mon. **73**(3), 299–301 (1966)

A Heuristic for the Traveling Salesperson Problem with Forbidden Neighborhoods on Regular 2D and 3D Grids

Philipp Armbrust, Philipp Hungerländer, and Anna Jellen

1 Introduction

In this work we present a heuristic approach for solving the Traveling Salesperson Problem with Forbidden Neighborhoods (TSPFN) on regular two- and three-dimensional grids. A visualization of regular grids is given in Fig. 1. The TSPFN (on regular grids) is an extension of the Traveling Salesperson Problem (TSP) that can be formally defined as follows.

Definition 1 (TSPFN on Regular Grids) Given are $mn\ell$ points $V = \{p^0, \ldots, p^{mn\ell-1}\}$ on the $m \times n \times \ell$ grid in the Euclidean space and a radius $r \in \mathbb{R}_0^+$. The task is to find a shortest Hamiltonian cycle $(p^{t_0}, p^{t_1}, \ldots, p^{t_{mn\ell-1}}, p^{t_0})$ with $mn\ell$ points, where $\| p^{t_i} - p^{t_{(i+1) \bmod mn\ell}} \| > r$ holds for all $i \in \{0, 1, \ldots, mn\ell - 1\}$.

For a detailed description of the main application of the TSPFN in mechanical engineering we refer to Kordaß [6]. Fischer and Hungerländer [3] determine closed-form TSPFN solutions on two dimensional regular grids for the smallest reasonable forbidden neighborhoods, i.e., $r \in \{0, 1, \sqrt{2}\}$ and furthermore introduce an Integer Linear Program (ILP) for determining optimal TSPFN solutions. An extension to three dimensions, as well as a study of closed-form solutions for the smallest reasonable forbidden neighborhoods, i.e., $r \in \{0, 1, \sqrt{2}, \sqrt{3}\}$, is provided in [4] and [5]. In [2] the authors investigate the TSPFN with forbidden neighborhood of radius 2, where an optimal solution equals a Knight's Tour if it exists.

A Knight's tour equals a Hamiltonian cycle where each step has length $\sqrt{5}$, i.e., is a Knight's move on a chessboard. Schwenk [7] characterized for which 2D grids Knight's tours exist and DeMaio and Mathew [1] extended these results for 3D

P. Armbrust (✉) · P. Hungerländer · A. Jellen
Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
e-mail: philipp.armbrust@aau.at; philipp.hungerlaender@aau.at; anna.jellen@aau.at

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_13

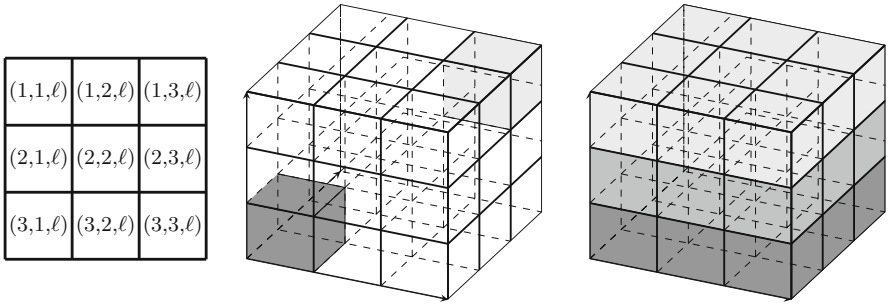


Fig. 1 Visualizations of the $3 \times 3 \times 3$ grid. The left picture shows the numbering of layers $\ell \in \{1, 2, 3\}$. In the middle cube, the dark gray cell has the coordinates $(3, 1, 1)$ and the light gray one has the coordinates $(1, 3, 3)$. In the right cube, each layer has one color, where layers with larger numbers are lighter

grids. Warnsdorff's Rule [9] is a heuristic that aims to find a Knight's path on an $m \times m$ chessboard.

Warnsdorff's Rule Choose a starting square arbitrarily. The $(n + 1)$th square of the path is chosen by considering the three following conditions:

- It is adjacent (reachable) to the nth square.
- It is unvisited (does not appear earlier in the path).
- It has the minimal number of adjacent, unvisited squares.

If there is more than one square that satisfies these conditions, the square is selected randomly. Since randomly choosing is usually not the best strategy, there are many investigations of so called tie-breaking rules, see e.g. Squirrel and Cull [8] for detailed studies.

This paper is structured as follows. We start with a description of our heuristic in Sect. 2. In Sect. 3 we conduct a computational study of the performance of the stated TSPFN heuristic. Finally, we conclude the paper and give suggestions for future work in Sect. 4.

2 Solution Approach

A heuristic approach for solving the TSPFN is of interest for different reasons. On the one hand, the TSPFN is well studied only for small forbidden neighborhoods, i.e., $r \leq 2$. To solve unknown instances exact approaches can be applied, i.e., Integer Linear Programs (ILPs), see in [3] and [5] respectively, but the performance on large grids is insufficient. On the other hand, the results of the heuristic are helpful for determining construction schemes and also lower bounds, i.e., in combination closed-form solutions, for further forbidden neighborhoods and grid dimensions. To our knowledge this is the first time a heuristic for the TSPFN is suggested, we depict it in detail in Algorithm 2.

Algorithm 2 Algorithmic description of our heuristic for the TSPFN

procedure COMPUTETSPFNTOUR($m \times n \times \ell$ grid, r)
Require: A regular $m \times n \times \ell$ grid and the radius of the forbidden neighborhood r.
Ensure: A feasible TSPFN tour with forbidden neighborhood r.
Initialize a variable h as shortest feasible step length $> r$, a variable s for the current visited square, an empty list C for the TSPFN tour, and an empty set S.
Choose a starting square, set s = the starting square, and add s to C.
while there exists an unvisited square **do**
Set S = FINDNEXTSQUARES($m \times n \times \ell$ grid, s, h, C).
Set s = CHOOSENEXTSQUARE($m \times n \times \ell$ grid, S, s, h, C).
Add s to C.
end while
return C
end procedure

procedure FINDNEXTSQUARES($m \times n \times \ell$ grid, s, h, C)
Require: A regular $m \times n \times \ell$ grid, the current visited square s, the preferred step length h, and the list of all visited squares C.
Ensure: A set S of possible reachable squares.
for each square v $\neq$ C **do**
if v is adjacent (reachable by a step of length h) to s and if v has the minimal number of adjacent, unvisited squares **then**
Add v to S.
end if
end for
return S
end procedure

procedure CHOOSENEXTSQUARE($m \times n \times \ell$ grid, S, s, h, C)
Require: A regular $m \times n \times \ell$ grid, a set of possible next squares S, the current visited square s, the preferred step length h, and the list of all visited squares C.
Ensure: A variable v that represents the square which should be chosen next.
if S is empty **then**
Initialize a variable j as shortest feasible step length >h and an empty set S.
Set S = FINDNEXTSQUARES($m \times n \times \ell$, s, j, C).
Set v = CHOOSENEXTSQUARE($m \times n \times \ell$, S, s, j, C).
else if S contains exactly one element **then**
Set v = this square.
else if S contains more than one element **then**
Apply a tie-breaking rule to determine v.
end if
return v
end procedure

Differences of our Heuristic to the original Warnsdorff's Rule

- The preferred step size depends on the forbidden neighborhood and is not fixed to $\sqrt{5}$.
- The step size is variable, i.e., steps of different sizes are allowed.

– The heuristic applies a tie-breaking rule instead of choosing the next square randomly. Tie-breaking rules have a big impact on the result of the heuristic because many decisions are taken by it. We apply the tie-breaking rule called *Maximal Euclidean Distance*: If there is more than one square that is optimal to choose next within the proposed algorithm, we choose the square with the maximal distance to the starting square. If there is more than one square with maximal distance, we choose one of the proposed squares randomly.
– It is mandatory to close the Hamiltonian path, otherwise the heuristic fails. Note that our tie-breaking rule aims to increase the chances of our heuristic to succeed.

3 Computational Results

In this section we outline results of our TSPFN heuristic. We implemented the heuristic in Java and ran computational results on a 2.7 GHz Intel Core i7 with 8 GB RAM, running Windows 10. We used Gurobi 8.0.0 as ILP solver.

The first experiment is designed to finetune critical input parameters, i.e., the number of runs and the best starting square. We ran tests on $n \times n$ and $n \times n \times n$ grids for $n \in \{6, 8, 10, 12, 14, 16\}$, for reasons of simplicity, and started the heuristic on each square 1000 times, i.e., $1000n^3$ runs. Squares on the corners and in the middle turn out to be good starting squares and 100 runs transpire to be enough to gain results of high-quality in a short time period. In the following experiments we therefore start the heuristic always for 100 times at the center square, i.e., square $\left(\left\lfloor \frac{n}{2} \right\rfloor, \left\lfloor \frac{n}{2} \right\rfloor\right)$.

In another experiment we consider the limitations of the ILP, i.e., what are the highest solvable dimensions. Already for grids with size greater or equal to 16×16, respectively $8 \times 8 \times 8$, and $r \geq \sqrt{2}$ a computation time of 10 min is not enough to compute at least a feasible solution. We summarize the results in Table 1.

Table 1 Results of the ILP for the TSPFN with different radii r of the forbidden neighborhood on $n \times n$ and $n \times n \times n$ grids, $n \in \{4, 6, 8, 10, 14, 16\}$

	$r = \sqrt{2}$		$r = \sqrt{5}$		$r = \sqrt{8}$		$r = 3$	
	Time	Gap	Time	Gap	Time	Gap	Time	Gap
6×6	< 1 s	0%	< 1 s	0%	300 s	0%	5 s	0%
10×10	64 s	0%	85 s	0%	TO	0.11%	TO	0.28%
14×14	562 s	0%	TO	0.12%	TO	0.11%	TO	2.26%
16×16	TO	–	TO	–	TO	–	TO	–
$4 \times 4 \times 4$	37 s	0%	TO	1.13%	1 s	0%	–	–
$6 \times 6 \times 6$	TO	–	TO	–	410 s	0%	TO	0.05%
$8 \times 8 \times 8$	TO	–	TO	–	TO	–	TO	–

The run time is given in seconds and the gap to optimality in percent. The time limit is set to 600 s

Table 2 Results of our heuristic for the TSPFN with different radii r of the forbidden neighborhood on $n \times n$ grids, $n \in \{8, 16, 74\}$

		$r = 0$	$r = 1$	$r = \sqrt{2}$	$r = 2$	$r = \sqrt{5}$	$r = \sqrt{8}$	$r = 3$
8×8	Sec	0.109	0.031	0.063	0.297	0.015	0.016	0.031
	Min	64.00	96.25	128.94	143.11	195.46	203.29	208.74
	Gap	0.00	0.60	0.00	0.00	6.17	4.99	2.69
	Opt	64.00	95.67	128.94	143.11	184.11	193.62	203.27
16×16	Sec	0.40	0.52	0.62	0.33	0.17	0.22	0.21
	Min	256.00	379.35	512.94	572.43	747.45	772.24	810.43
	Gap	0.00	2.00	0.00	0.00	–	–	–
	Opt	256.00	371.88	512.94	572.43	–	–	–
74×74	Sec	61.80	64.37	62.67	74.95	63.34	62.73	73.35
	Min	5476.00	7827.46	10,953.77	12,244.70	15,606.97	16,431.30	17,353.47
	Gap	0.00	0.50	0.01	0.00	–	–	–
	Opt	5476.00	7788.05	10,952.94	12,244.70	–	–	–

Runtime for 100 starts, calculated minimal lengths, gap to optimality in percent, and optimal lengths are depicted

Next we test our TSPFN heuristic on grids up to size 74×74, respectively $18 \times 18 \times 18$. We observe that the lengths of the tours produced are close to optimum, in case the optimum is known. The computation time is still underneath a second for grids where the ILP fails to produce a solution within 10 min. The heuristic delivers results on the largest considered grids in approximate 1 min. The optimal solutions are taken from [3–5]. In Tables 2 and 3 we summarize the results obtained.

4 Conclusion and Outlook

In this paper we proposed a heuristic to solve the Traveling Salesperson Problem with Forbidden Neighborhoods (TSPFN) on regular two- and three-dimensional grids for arbitrary radii of the forbidden neighborhood. The TSPFN heuristic is based on the well-known heuristic called Warnsdorff's Rule that aims to find a Knight's path on a chessboard. We conducted a computational study to show that our TSPFN heuristic delivers high-quality solutions, i.e., optimum or near-optimum TSPFN tours. We also showed that the heuristic scales very well, even for large grids and radii. For future work it remains to determine further closed-form solutions for the TSPFN. Additionally we want to extend our heuristic to consider the case that the forbidden neighborhood is time dependent and hence can change over time or be active for more than one step. Finally it would be interesting to examine the TSPFN on grids that are not complete, i.e., there are cells in the grid that cannot be visited.

Table 3 Results of our heuristic for the TSPFN with different radii r of the forbidden neighborhood on $n \times n \times n$ grids, $n \in \{6, 10, 18\}$

		$r = 0$	$r = 1$	$r = \sqrt{2}$	$r = \sqrt{3}$	$r = 2$	$r = \sqrt{5}$	$r = \sqrt{6}$	$r = \sqrt{8}$	$r = 3$
$6 \times 6 \times 6$	Sec	0.36	0.28	0.25	0.27	0.29	0.18	0.14	0.20	0.15
	Min	216.00	306.11	389.13	435.17	482.99	530.19	621.33	648.00	649.45
	Gap	0.00	0.00	–	0.29	0.00	–	–	–	–
	Opt	216.00	306.11	–	433.89	482.99	–	–	–	–
$10 \times 10 \times 10$	Sec	2.74	2.70	2.375	2.26	3.62	3.38	2.58	3.91	0.429
	Min	1000.00	1414.85	1762.98	2004.18	2236.07	2451.30	2834.17	3000.00	3162.58
	Gap	0.00	0.00	–	0.11	0.00	–	–	–	–
	Opt	1000.00	1414.85	–	2001.88	2236.07	–	–	–	–
$18 \times 18 \times 18$	Sec	75.31	113.23	102.77	85.72	157.70	152.77	103.19	174.47	151.77
	Min	5832.83	8248.32	10,217.43	11,670.80	13,040.75	14,286.52	16,498.13	17,496.00	18,442.71
	Gap	0.01	0.00	–	0.04	0.00	–	–	–	–
	Opt	5832.00	8248.32	–	11,665.88	13,040.75	–	–	–	–

Runtime for 100 starts, calculated minimal lengths, gap to optimality in percent, and optimal lengths are depicted

References

1. DeMaio, J., Mathew, B.: Which chessboards have a closed knight's tour within the rectangular prism? Electron. J. Comb. **18**, #P8 (2011)
2. Firstein, M., Fischer, A., Hungerländer, P.: Closed almost knight's tours on 2D and 3D chessboards. In: Operations Research Proceedings. Springer, Cham (2017)
3. Fischer, A., Hungerländer, P.: The traveling salesman problem on grids with forbidden neighborhoods. J. Comb. Optim. **34**(3), 891–915 (2017)
4. Fischer, A., Hungerländer, P., Jellen, A.: The traveling salesperson problem with forbidden neighborhoods on regular 3D grids. In: Operations Research Proceedings. Springer, Cham (2017)
5. Jellen, A.: The traveling salesperson problem with forbidden neighborhoods on regular 3D grids. Master thesis, Alpen-Adria-Universität Klagenfurt (2018)
6. Kordaß, R.: Untersuchungen zum Eigenspannungs- und Verzugsverhalten beim Laserstrahlschmelzen. Master thesis, Technische Universität Chemnitz (2014)
7. Schwenk, A.J.: Which rectangular chessboards have a knight's tour? Math. Mag. **64**(5), 325–332 (1991)
8. Squirrel, D. Cull, P. A Warnsdorff-rule algorithm for knight's tours on square chessboards. Oregon State REU Program (1996)
9. Warnsdorff, H.C.: Des Rösselsprungs einfachste und allgemeinste Lösung, Schmalkalden (1823)

A Mixed Integer Linear Program for Optimizing the Utilization of Locomotives with Maintenance Constraints

Sarah Frisch, Philipp Hungerländer, Anna Jellen, and Dominic Weinberger

1 Introduction

This paper discusses a variant of the Locomotive Scheduling Problem (LSP), where the aim is to assign a fleet of locomotives to a set of scheduled trains, such that the overall costs are minimized. The rolling stock represents one of the main costs of a rail transportation system. The costs are composed of the one-time investment for the acquisition and running costs for fuel and maintenances. Our aim is to reduce these costs by minimizing either the number of locomotives used, the number of deadhead kilometers, or the combination of both. After giving an overview of already existing approaches in locomotive scheduling in Sect. 2, we state our specific problem and model it on a sparse weighted directed multigraph in Sect. 3. We continue with formulating a Mixed Integer Linear Program (MILP) based on the problem graph in Sect. 4. Finally, we conduct a computational study on a part of the Austrian railway network in Sect. 5. The aim of our proposed optimization model is to test the different objective functions relating to runtime and quality of the solution. We further test our MILP within a rolling horizon approach. We show that our MILP delivers high-quality solutions for the LSP in freight transportation and is applicable by the Rail Cargo Austria (RCA).

S. Frisch (✉) · P. Hungerländer · A. Jellen · D. Weinberger
Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
e-mail: sarah.frisch@aau.at; philipp.hungerlaender@aau.at; anna.jellen@aau.at; dominic.weinberger@hex-solutions.com

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_14

2 Literature Review

Optimization in rail transportation is a widely studied field. Vaidyanathan et al. [5] formulate the LSP as an integer-programming string decomposition problem on a suitably defined space-time network considering fueling and maintenances. They solve it by an aggregation-disaggregation based algorithm. Noori and Ghannadpour [4] consider maintenances but in contrast to our approach the trains in their model have different priorities to be conducted. They model a Vehicle Routing Problem with Time Windows, where the trains act as costumers, and solve it heuristically. Reuther et al. suggest several algorithms to solve the rolling stock rotation problem, i.e., the LSP, where a cyclic planning horizon over 1 week due to a periodic input schedule is considered. They state a hypergraph based Integer Program (IP) formulation in [1], which is extended for considering maintenances in [2]. They solve the IP by using an algorithm based on column generation and a method called rapid branching. In [3] they provide a local search heuristic to solve large scale instances for the DB Fernverkehr AG and ICE vehicles. In contrast to [1] where hyperarcs are introduced, we define a sparse multigraph and based on this graph we propose a MILP that we solve without applying heuristics. We minimize over deadhead kilometers and locomotives used, and provide three objective functions. This differs from [5] where costs for maintenances and ownership are minimized, and from [1] where costs for locomotives, services, deadhead kilometers, regularity, and coupling are minimized. Compared to the approaches mentioned above our MILP is based on a sparse multigraph, where it is not necessary to model the time component explicitly. Our MILP is applicable on variable calender dates and within a rolling horizon approach.

3 Problem Description and Modeling

In this section we formulate our problem by defining a graph that will later build the basis for our MILP. We assume to have m locomotives $F = \{f_1, \ldots, f_m\}$ at disposal. Each $k \in F$ is allowed to travel $r_k^{\max} \in \mathbb{R}^+$ kilometers until a maintenance must be conducted. We have a schedule of trains that must be performed with information about stations and time for departure and arrival. We introduce a directed, weighted multigraph $D = (V, A_1, A_2, A_3)$, where V is the set of nodes and A_1, A_2, and A_3 are sets of edges. Nodes represent stations at a particular time, i.e., Station 1 at 10:00 am and Station 1 at 12:00 pm are represented by two different nodes. Edges display all kinds of trips that a locomotive is able to perform, A_1 represents scheduled trips, A_2 represents deadhead trips, and A_3 represents deadhead trips including a maintenance. We define two nodes for each scheduled trip: a *departure node* $v \in V^d \subset V$ and an *arrival node* $v \in V^a \subset V$.

For each scheduled trip we draw an edge adjacent to the corresponding departure and arrival nodes. Furthermore, we connect edges of scheduled trips by edges that

represent deadhead trips with or without maintenances. The allocation of locomotives can be illustrated as follows: Locomotives travel alternating on scheduled trips and deadhead trips with or without maintenances, such that each node (station at a specific time) is visited exactly once by exactly one locomotive.

In order to meet all constraints, we enlarge the set of nodes and refine the definitions of the graph. At the beginning of the planning horizon each locomotive has mileage r_k^s and is placed at its *start node* $v_k \in V^s \subset V$, a station at the time of the beginning of the planning horizon. We also introduce an artificial *final node* v_f and define the set $\{v_f\} =: V^f \subset V$. We draw an edge for a deadhead trip with distance zero, from each arrival node to the final node. In this way we are able to model (in our MILP) that all locomotives can end in an arbitrary station. In summary the set of nodes is $V = V^d \cup V^a \cup V^s \cup V^f$. Now let us also give a more detailed description of the sets of edges:

(a) *The set of edges for scheduled trips* A_1*:* For each scheduled trip we define a directed edge $(i, j) \in A_1 \subseteq V \times V$ which is weighted by the distance d_{ij} between the stations related to the corresponding departure and arrival nodes. In order to provide enough power for pulling the trains, only some locomotives are allowed to conduct certain trains. Hence we introduce the set $P_{ij} = \{k \in F : k \text{ is permitted to conduct the trip from } i \text{ to } j\}$.

(b) *The set of edges for deadhead trips* A_2*:* For a possible deadhead trip we introduce a directed edge $(i, j) \in A_2 \subseteq V \times V$ that is weighted by the distance d_{ij} between the stations related to Nodes i and j. We call Node j *reachable* for a locomotive starting in Node i, if there exists a path between the corresponding stations and if the time at i plus the travel duration is smaller than the time at j. We draw such edges between an arrival and a reachable departure node, a start node and each reachable departure node, and each start, respectively arrival, node and the final node.

(c) *The set of edges for deadhead trips with maintenances* A_3*:* For a deadhead trip with maintenance we define a directed edge $(i, j) \subset V \times V$. As for edges in A_1 we have a set P_{ij} that contains all locomotives that are permitted to conduct the trip from i to j. For edges of deadhead trips with maintenances P_{ij} contains exactly one element $k \in F$. Therefore we obtain edges of the form $(i, j, k) \in A_3 \subseteq V \times V \times F$, that are weighted by the distance d_{ij} between the corresponding stations of Nodes i and j. We draw such an edge from i to j for a $k \in F$, if all conditions for an edge in A_2 are satisfied, and if additionally the time at i plus the travel duration plus the time for the maintenance of k is smaller than the time at j, and k is admitted to be maintained in j.

Finally let us introduce a subset of nodes that is needed for the MILP formulation: $V_k^{s+1} \subset V$ contains all nodes $v \in V \setminus V^f$ such that there is an edge in A_2 or A_3 connecting the start station v_k of $k \in F$ and v. In total we obtain a sparse graph with the advantage that for the corresponding MILP it is not necessary to introduce variables for explicitly modeling the time component of the trains.

4 Mathematical Formulation

Based on the described Graph D we formulate a Mixed Integer Linear Program (MILP) with different objective functions. We start with minimizing the overall deadhead kilometers and introduce the following decision variables:

- $x_{ij} \in \{0, 1\}$, $(i, j) \in A_2$, with $x_{ij} = 1$, if edge (i, j) is used.
- $y_{ijk} \in \{0, 1\}$, $(i, j, k) \in A_3$, with $y_{ijk} = 1$, if edge (i, j, k) is used.
- $q_{ik} \in \{0, 1\}$, $i \in V \setminus V^f$, $k \in F$, with $q_{ik} = 1$, if Node i is visited by Locomotive k.
- $r_i \in \mathbb{R}^+$, $i \in V$, denotes the mileage in Node i.

$$\min \quad \sum_{(i,j)\in A_2} d_{ij}x_{ij} + \sum_{(i,j,k)\in A_3} d_{ijk}y_{ijk} \qquad \text{(Objective I)}$$

$$\text{s.t.} \quad \sum_{(i,j)\in A_2} x_{ij} + \sum_{(i,j,k)\in A_3} y_{ijk} = 1, \quad j \in V^d, \tag{1}$$

$$\sum_{(i,j)\in A_2} x_{ij} + \sum_{(i,j,k)\in A_3} y_{ijk} = 1, \quad i \in V^a \cup V^s, \tag{2}$$

$$\sum_{(i,j)\in A_2} x_{ij} = m, \quad j \in V^f, \tag{3}$$

$$q_{ik} = 1, \quad i \in V^s,\ k \in F,\ i = v_k, \tag{4}$$

$$\sum_{k\in F} q_{ik} = 1, \quad i \in V \setminus V^f, \tag{5}$$

$$\left.\begin{aligned} &q_{ik} = q_{jk}, \quad (i, j) \in A_1,\ k \in F, \text{if } k \in P_{ij}, \\ &q_{ik} + q_{jk} = 0, \quad (i, j) \in A_1,\ k \in F, \text{if } k \notin P_{ij}, \end{aligned}\right\} \tag{6}$$

$$q_{jk} \geq q_{ik} - (1 - x_{ij}), \quad (i, j) \in A_2,\ k \in F,\ j \notin V^f, \tag{7}$$

$$y_{ijk} \leq q_{\ell k}, \quad (i, j, k) \in A_3,\ \ell \in \{i, j\}, \tag{8}$$

$$r_i = r_i^s, \quad i \in V^s, \tag{9}$$

$$r_j = r_i + d_{ij}, \quad (i, j) \in A_1, \tag{10}$$

$$r_j \geq r_i + d_{ij} - M_1(1 - x_{ij}), \quad (i, j) \in A_2, \tag{11}$$

$$r_j \leq M_2(1 - y_{ijk}), \quad (i, j, k) \in A_3, \tag{12}$$

$$r_i \leq r_k^{\max} + M_3(1 - q_{ik}), \quad i \in V \setminus V^f,\ k \in F, \tag{13}$$

$$r_i + d_{ijk} \leq r_k^{\max} + M_3(1 - y_{ijk}), \quad (i, j, k) \in A_3, \tag{14}$$

$$x_{ij} \in \{0, 1\}, \quad (i, j) \in A_2, \tag{15}$$

$$y_{ijk} \in \{0,1\}, \quad (i,j,k) \in A_3, \tag{16}$$

$$q_{ik} \in \{0,1\}, \quad i \in V \setminus V^f,\ k \in F, \tag{17}$$

$$r_i \geq 0, \quad i \in V. \tag{18}$$

Constraints (1) and (2) guarantee that each departure node has exactly one outgoing edge and each arrival respectively start node has exactly one ingoing edge. Equation (3) secure that all locomotives enter the final node. Constraints (4) link locomotives with their departure stations. Equation (5) ensure that except for the final station every station is visited by exactly one locomotive. Constraints (6) guarantee that only permitted locomotives conduct scheduled trips. Inequalities (7) ensure for $(i, j) \in A_2$ that Nodes i and j are visited by the same locomotive. Constraints (8) secure that $(i, j, k) \in A_3$ can be used for a locomotive just if it has visited the corresponding departure and arrival node. The mileages are initialized and updated in Constraints (9) and (10). The mileage of deadhead kilometers is updated by Inequalities (11). The mileages after conducted maintenances are initialized by Constraints (12). Inequalities (13) and (14) avoid to exceed the maximal permitted mileage of a locomotive until the next maintenance. We recommend to choose $M_1 = \max_{k\in F} r_k^{\max} + \max_{(i,j)\in A_2} d_{ij}$, $M_2 = \max_{k\in F} r_k^{\max}$, and $M_3 = \max_{(i,j,k)\in A_3} d_{ijk}$.

For minimizing over the number of locomotives we introduce additional decision variables $s_k \in \{0,1\}$, $k \in F$, with $s_k = 1$, if Locomotive k is used at least once within the considered planning horizon. We alter the MILP by replacing (Objective I) by (Objective II) and adding Constraints (19).

$$\min \quad \sum_{k\in F} s_k \tag{Objective II}$$

$$\text{s.t.} \quad s_k \geq \left(\sum_{j\in V_k^{s+1}} x_{ij} + y_{ijk} \right), \quad i \in V^s,\ k \in F. \tag{19}$$

Finally we suggest to combine the two proposed objective functions. To do so we run our two MILPs and use their objective values, denoted by z_{km} and z_{loc}, as weights to gain a good trade-off. We alter our second MILP by replacing (Objective II) by (Objective III):

$$\min \quad z_{loc}\left(\sum_{(i,j)\in A_2} d_{ij} x_{ij} + \sum_{(i,j,k)\in A_3} d_{ijk} y_{ijk} \right) + z_{km} \sum_{k\in F} s_k \tag{Objective III}$$

5 Computational Study

To simulate a real world scenario regarding the RCA's daily operations we tested our three MILPs on a part of the Austrian railway network. We created train schedules regarding an approximated traffic density and used real-world parameters for the specific dates and durations of maintenances. Considering a planning horizon of a week we solved each problem entirely, and then applied our MILPs within a rolling horizon approach by splitting the time interval into half. Our experimental results are summarized in Table 1.

We observe that the more locomotives are at disposal, the less deadhead kilometers are driven. But the CPU time increases due to the sharply rising number of edges in A_3. Considering the whole time interval the corresponding MILPs deliver high quality solutions, but they are too time consuming for large instances. By splitting the interval we achieve excellent solution times and an acceptable solution quality. Results under (Objective III) show a notably good trade-off between the number of locomotives used and the number of deadhead kilometers.

As future work we intend to reduce Sets A_2 and A_3 and thus the variables of our MILP by permitting corresponding edges just within a certain time window. The

Table 1 We display the instance and model size, the solution time (mm:ss), the number of locomotives used (LU), and the number of deadhead kilometers (DH) of the MILP under each proposed objective

Instance				Whole time interval			Split time interval		
#Locs/#trains	$\#x_{ij}$	$\#y_{ijk}$	Obj	CPU time	LU	DH	CPU time	LU	DH
40/80	7159	170,934	I	04:03	34	47.77	00:08	31	52.57
			II	21:27	20	268.08	00:18	27	313.57
			III	(29:06) 03:36	24	51.37	(00:35) 00:09	27	61.62
40/120	12,301	372,967	I	18:18	39	68.91	00:25	36	75.27
			II	52:48	29	387.80	00:47	36	474.01
			III	(01:23:34) 12:28	32	70.53	(01:42) 00:30	36	80.34
60/60	6335	145,015	I	02:11	38	36.16	00:11	37	40.06
			II	28:32	15	193.52	00:02	21	203.65
			III	(34:50) 04:07	19	40.75	(00:26) 00:13	26	45.90
60/80	8998	256,389	I	12:07	45	46.15	00:14	41	50.17
			II	14:33	20	257.18	00:31	33	299.80
			III	(34:54) 08:14	24	49.75	(00:51) 00:06	32	56.53
60/120	15,050	559,440	I	39:57	50	46.57	00:57	50	60.32
			II	TO	–	–	03:54	41	452.65
			III	–	–	–	(05:56) 01:05	42	66.41

The time within the braces displays the overall solution time of all three MILPs. The time out (TO) per MILP was set to an hour. All experiments were performed using Gurobi 7.5.2 on a Linux 64-bit machine equipped with Intel(R) Xeon(R) CPU e5-2630 v3@2.40 GHz and 128 GB RAM in multi thread mode with four cores

proposed model could be expanded by capacities of maintenance and train sections. Minimizing the overall number of maintenances conducted could be considered as another objective.

References

1. Borndörfer, R., Reuther, M., Schlechte, T., Weider, S.: A hypergraph model for railway vehicle rotation planning. In: 11th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (2011)
2. Borndörfer, R., Reuther, M., Schlechte, T., Weider, S., Vehicle rotation planning for intercity railways. In: Proceedings of Conference on Advanced Systems for Public Transport (2012)
3. Borndörfer, R., Reuther, M., Schlechte, T., Waas, K., Weider, S.: Integrated optimization of rolling stock rotations for intercity railways. Transp. Sci. **50**(3), 863–877 (2016)
4. Noori, S., Ghannadpour, S.F.: Locomotive assignment problem with train precedence using genetic algorithm. J. Ind. Eng. Int. **8**, 9 (2012)
5. Vaidyanathan, B., Ahuja, R.K., Orlin, J.B.: The locomotive routing problem. Transp. Sci. **42**(4), 492–507 (2008)

Proportional Apportionment for Connected Coalitions

Sebastian Goderbauer and Leonie Ermert

1 Introduction

In electoral systems, which are based on proportional representation, each political party is represented in parliament in proportion to the number of people who vote for it. To translate large vote counts of those to be represented into small numbers of parliamentary seats, procedures called *apportionment methods* are employed. Since a total number of seats is usually prespecified by law and must be dealt out precisely, simply rounding is not sufficient. Apportionment methods are able to meet this requirement and, more importantly, ensure proportionality.

In Germany, an apportionment method is not only used after a federal election, but also in its preparation: A prespecified number of electoral districts is apportioned among the 16 German federal states in proportion to their population figures. After that, electoral districts are designed in each state separately. The task of partitioning a geographical territory into a given number of electoral districts is called *political districting problem* and includes different constraints and (optimization) criteria. The fact that an apportionment method is part of the very first step in the design process of electoral districts in, e.g., Germany, leads to the basic motivation of our work: Development of generalized apportionment methods that are applicable in further steps of the (re)districting process.

S. Goderbauer (✉)
Lehrstuhl für Operations Research, RWTH Aachen University, Aachen, Germany

Lehrstuhl II für Mathematik, RWTH Aachen University, Aachen, Germany
e-mail: goderbauer@or.rwth-aachen.de

L. Ermert
Lehrstuhl für Operations Research, RWTH Aachen University, Aachen, Germany
e-mail: leonie.ermert@rwth-aachen.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_15

Contribution Classical apportionment methods assign one integer to each *recipient*. We generalize a prime class of apportionment methods by allowing *coalitions* of recipients: The latter can be grouped in order to improve proportionality of the apportionment. We show that this additional combinatorics leads to an *NP-hard* problem. In contrast, the standard setting can be solved in linear time. We propose a *mixed-integer linear program* (MILP) for the generalization, discuss extensions, and point out its application in the *political districting problem*.

Related Work Apportionment methods were studied by Balinski and Young [2], Kopfermann [10], and Pukelsheim [11]. The political districting problem is discussed in surveys of Goderbauer et al. [8] and Ricca et al. [12]. The idea to generalize apportionment methods was triggered by work of Goderbauer [5, 6].

2 Apportionment Problem and Divisor Methods

The general setting of an apportionment problem reads as follows. For a vector $v \in \mathbb{R}^n$, $n \in \mathbb{N}$ we define the sum of its components as $v_+ := \sum_{i=1}^{n} v_i$.

Definition 1 (Apportionment Problem) Given *target size* $h \in \mathbb{N}_{\geq 1}$, number of *recipients* $\ell \in \mathbb{N}_{\geq 2}$, and *weight vector* $0^\ell \neq w = (w_1, \ldots, w_\ell) \in \mathbb{Q}^\ell_{\geq 0}$. An *apportionment problem* asks for an *apportionment vector* $x = (x_1, \ldots, x_\ell) \in \mathbb{N}^\ell_{\geq 0}$ with $x_+ = h$, i.e., an allocation x_i for each recipient i so that target size is met.

An apportionment problem does not contain any requirement on proportionality between w and x since there exists no unique measurement of disproportionality. Certainly, of interest are apportionment vectors x with $\frac{x_i}{h} \approx \frac{w_i}{w_+}$ for all i. Equality in all equations is called *perfect proportionality*. Minimizing the deviation from perfect proportionality is the purpose of suitable solution methods.

To define solution methods for the apportionment problem, some notation is introduced. For target size $h \in \mathbb{N}_{\geq 1}$ and weight vector $w \in W := \bigcup_{\ell \geq 2} \mathbb{Q}^\ell_{\geq 0} \setminus \{0^\ell\}$, denote the dimension of w as $\ell(w)$ and define the set of all feasible apportionment vectors as $\mathbb{N}^{\ell(w)}(h) := \{x \in \mathbb{N}^{\ell(w)}_{\geq 0} : x_+ = h\} \in X := \bigcup_{h \geq 1} \bigcup_{\ell \geq 2} \mathbb{N}^\ell(h)$.

Definition 2 (Apportionment Method) A mapping $A : \mathbb{N}_{\geq 1} \times W \to X$ with $\emptyset \neq A(h, w) \subseteq \mathbb{N}^{\ell(w)}(h)$, $h \in \mathbb{N}_{\geq 1}$, $w \in W$ is an *apportionment method*.

In the following, we consider the "most powerful apportionment methods" [11, p. 72]: *divisor methods*. At its core, divisor methods use a flexible divisor to scale the weights to interim quotients of an appropriate order of magnitude and round these to integers using a specific rule. We focus on the *divisor method with standard rounding*, but our findings can be transferred to other divisor methods.

Definition 3 (Rule of Standard Rounding) The *rounding rule of standard rounding* $[\![\cdot]\!]$ is defined for $t \in [0, \infty)$ and $n \in \mathbb{N}_{\geq 0}$ with $[\![t]\!] := \{0\}$ if $t = 0$ and

$$[\![t]\!] := \begin{cases} \{n\} & \text{if } t \in \left(n - \frac{1}{2}, n + \frac{1}{2}\right), \\ \{n-1, n\} & \text{if } t = n - \frac{1}{2} > 0. \end{cases}$$

Lemma 1 *For all* $t \in [0, \infty)$ *and* $n \in \mathbb{N}_{\geq 0}$ *holds:* $n \in [\![t]\!] \Leftrightarrow n - \frac{1}{2} \leq t \leq n + \frac{1}{2}$.

Lemma 1 is a direct consequence of Definition 3. The divisor method induced by the rule of standard rounding is denoted with *DivStd* and defined as follows.

Definition 4 (DivStd) The *divisor method with standard rounding* reads $\mathrm{DivStd}(h, w) := \left\{ x \in \mathbb{N}^{\ell}(h) : x_1 \in \left[\!\left[\frac{w_1}{D} \right]\!\right], \ldots, x_\ell \in \left[\!\left[\frac{w_\ell}{D} \right]\!\right] \text{ for some } D > 0 \right\}$.

The solution set of divisor methods, including DivStd, is easily computable. In fact, after a sensible initialization of divisor D, it needs linear many iterations to meet the target size with the summed up rounded quotients [11].

Theorem 1 $\mathrm{DivStd}(h, w)$ *is computable in* $\mathcal{O}(\ell(w))$ *for* $h \in \mathbb{N}_{\geq 1}$ *and* $w \in W$.

In 1910, Sainte-Laguë [13] proved the following optimality characteristic [11].

Theorem 2 *It holds:* $x \in \mathrm{DivStd}(h, w) \Leftrightarrow x \in \underset{y \in \mathbb{N}^{\ell(w)}(h)}{\arg\min} \sum_{i=1}^{\ell(w)} w_i \cdot \left(\frac{y_i / h}{w_i / w_+} - 1 \right)^2$.

3 Proportional Apportionment for Coalitions

A *coalition* is a non-empty set of recipients. In the literature, it is analyzed which apportionment method encourages coalitions or, quite the opposite, encourages schisms [2]. Coalitions are encouraged, if the expected success rate, e.g., number of seats, is higher if recipients form a coalition before allocation instead of standing alone. So far, the concept of coalitions has not been part of an apportionment problem itself. In our generalization, forming coalitions gets part of the decision to be made. In the classical setting, these coalitions are fixed to singletons. The following theorem guarantees that the formation of coalitions in our generalization of DivStd does not lead to (dis)advantages for recipients.

Theorem 3 ([1]) DivStd *is the unique divisor method that encourages neither coalitions nor schisms.*

In applications, not every coalition may be desired. Therefore, we consider a graph $G = (V, E)$ with set of recipients $V = \{1, \ldots, \ell\}$ as input of the generalized problem and request that the *subgraph induced by* each coalition $C \subset V$, denoted with $G[C] := (C, \{(i_1, i_2) \in E : i_1, i_2 \in C\})$, has to be connected.

3.1 Generalized Apportionment Problem and Divisor Methods

Our generalized apportionment problem for connected coalitions reads as follows.

Definition 5 (Apportionment Prob. for Connected Coalitions (APCC)) Given $h \in \mathbb{N}_{\geq 1}$, $w \in W$, an apportionment method A, and a graph $G = (V, E)$ with $V = \{1, \ldots, \ell(w)\}$. An *apportionment problem for connected coalitions (APCC)* asks for a partition of V in $2 \leq p \leq \ell(w)$ coalitions $\mathcal{C} = \{C_1, \ldots, C_p\}$ with $G[C_j]$ connected for all j, and an apportionment vector $x \in A(h, w^{\mathcal{C}})$ with coalition weight $w_j^{\mathcal{C}} := \sum_{i \in C_j} w_i$ for $j = 1, \ldots, p$.

We exclude $p = 1$ since the trivial partition $\mathcal{C} = \{V\}$ has always perfect proportionality. Let $\mathcal{P}_{\geq 2}(V)$ be the set of all non-trivial partitions of V. To evaluate a solution of APCC, i.e., coalitions $\mathcal{C}$ (with its apportionment $x \in A(h, w^{\mathcal{C}})$), a function $f : \mathcal{P}_{\geq 2}(V) \times X \to \mathbb{R}_{\geq 0}$ can be employed. Such an f is called *APCC objective* if a value of f equals zero if and only if the evaluated apportionment has perfect proportionality. DivStd's measurement of disproportionality (Theorem 2) fulfills this requirement. However, applications may require a different objective.

Definition 6 (DivCoalStd) Given an APCC instance with apportionment method DivStd and an APCC objective f. We define *DivStd for connected coalitions under objective f (DivCoalStd^f)* with the following characterization:
$(\mathcal{C}, x) \in \text{DivCoalStd}^f(h, w, G) :\Longleftrightarrow \mathcal{C} = \{C_1, \ldots, C_p\}$, x optimal solution of

$$
\begin{aligned}
&\min\ f(\mathcal{C}, x) \\
&\text{s.t. } p \in \mathbb{N}, 2 \leq p \leq \ell \\
&\quad \mathcal{C} = \{C_1, \ldots, C_p\} \text{ partition of } V \text{ with } C_j \neq \emptyset \quad \forall j = 1, \ldots, p \\
&\quad G[C_j] \text{ connected} \quad \forall j = 1, \ldots, p \\
&\quad x \in \text{DivStd}(h, w^{\mathcal{C}})
\end{aligned}
\tag{1}
$$

3.2 Complexity

Computing DivStd takes linear time (cf. Theorem 1). We show that DivCoalStd is more complex. This is not only due to the required connectedness of the coalitions, which itself is NP-hard. Let K_ℓ be the complete graph on ℓ nodes.

Theorem 4 ([3], Problem No. 6 in [9]) *Partitioning a graph into connected subgraphs of bounded size or weight is NP-hard.*

Theorem 5 *Computing* DivCoalStd$^f(h, w, G)$ *is NP-hard even if G is complete.*

Proof Reduction from PARTITION. Let $U = \{1, \ldots, \ell\}$, $\ell \in \mathbb{N}_{\geq 2}$ with $s_i \in \mathbb{N}_{\geq 1}$, $i \in U$ be given. Prove: There is $U' \subseteq U$ with $\sum_{i \in U'} s_i = \sum_{i \in U \setminus U'} s_i$ iff

DivCoalStd$^f(h, w, K_\ell)$ with $h := 2$, $w_i := 2 \cdot \frac{s_i}{s_+}\ \forall i = 1, \ldots, \ell$ (note, $w_+ = h$) has a solution with perfect proportionality, i.e., objective value 0 in (1) for all f.
$\Rightarrow$: $C_1 := U', C_2 := U \setminus U'$ fulfills $\sum_{i \in C_1} w_i = \sum_{i \in C_2} w_i = 1$. Choose $x := (1, 1)$.
$\Leftarrow$: Since $w_+ = h$, perfect proportionality is equivalent to $x_j = \sum_{i \in C_j} w_i\ \forall j$ and therefore to $\sum_{i \in C_j} w_i \in \mathbb{N}\ \forall i$. Since $\mathbb{N}$ is closed under addition, coalitions can be merged: W.l.o.g. assume $p = 2$. Since $w_i > 0\ \forall i$, we get $x_1 = \sum_{i \in C_1} w_i = 1 = \sum_{i \in C_2} w_i = x_2$, implying $\sum_{i \in C_1} s_i = \sum_{i \in C_2} s_i$. Choose $U' := C_1$. □

3.3 Mixed-Integer Linear Programming (MILP) Formulation

DivStd can be formulated as integer points of a linearly described region.

Lemma 2 $$\text{DivStd}(h, w) = \left\{ (x_1, \ldots, x_{\ell(w)}) : \begin{array}{l} \sum_i x_i = h,\ x_i \in \mathbb{N}_{\geq 0}\ \forall i, \\ -\frac{1}{2} \leq w_i \mu - x_i \leq \frac{1}{2}\ \forall i, \mu \geq 0 \end{array} \right\}.$$

Proof With $D := 1/\mu$, Lemma 1 reads $x_i \in [\![w_i \mu]\!] \Leftrightarrow -\frac{1}{2} \leq w_i \mu - x_i \leq \frac{1}{2}$. Note, $\mu = 0$ is only feasible on the right for $h = 0$ and this is not allowed as input. □

In the following, we propose a formulation for problem (1). Let $B = \{1, \ldots, \ell\}$ be coalitions' indices. Consider variables: Binary $\delta_{i,b}$ equals 1 iff coalition $b \in B$ contains recipient $i \in V$. Binary γ_b equals 1 iff coalition $b \in B$ is non-empty. Integer x_b denotes allocation for $b \in B$. A formulation of all feasible points reads:

$$\begin{array}{lll} \sum_{b \in B} x_b = h, \quad \sum_{b \in B} \gamma_b \geq 2 & & \\ \sum_{b \in B} \delta_{i,b} = 1 \quad \forall i \in V & & \mu \geq 0 \\ -\frac{1}{2} \leq \left(\sum_{i \in V} w_i \delta_{i,b} \right) \cdot \mu - x_b \leq \frac{1}{2} \quad \forall b \in B & & x_b \in \mathbb{N}_{\geq 0} \quad \forall b \in B \\ G[\{i \in V : \delta_{i,b} = 1\}] \text{ connected} \quad \forall b \in B & & \gamma_b \in \{0, 1\} \quad \forall b \in B \\ \delta_{i,b} \leq \gamma_b \leq \sum_{i \in V} \delta_{i,b} \quad \forall i \in V, b \in B & & \delta_{i,b} \in \{0, 1\} \quad \forall i \in V, b \in B \end{array} \tag{2}$$

Theorem 6 *If APCC objective f can be modeled linearly, problem (1) and therefore* DivCoalStd^f *can be formulated as an MILP.*

Proof Consider formulation (2). Products $\delta_{i,b} \cdot \mu$ can be linearized by introducing a new variable and additional constraints [folklore] since $\mu \leq \frac{h + \frac{1}{2}}{\max_i w_i}$ holds. Connectedness of subgraphs can be ensured by, e.g., *separator inequalities* [14]. □

4 Application: Political Districting Problem

For the design of electoral districts, German law stipulates that boundaries of rural and urban districts (a level of administrative subdivisions) should be respected where possible [8]. It is preferred that a boundary of an electoral district matches a boundary of a rural/urban district. This guideline, called *administrative (adm.) conformity*, is an important objective in German political districting (PD) practice [7]. Another legal aim requests for preferably equal population in each electoral district, i.e., small electoral district's *population deviation*.

We utilize our results to allocate a federal state's number of electoral districts among connected coalitions of rural and urban districts. This APCC results in a number of smaller PD instances, one from each non-empty connected coalition C_b. Each resulting instance has (1) a modest number of electoral districts, (2) adm. conformity for at least the instance's external borders, and (3) small population deviation. Coalitions C_b with $x_b = 1$ lead to already solved PD instances with perfect adm. conformity. Remaining instances can be solved with PD algorithms [8, 12]. As an example, we examine the federal state of Hesse.

APCC Instance for DivCoalStd Hesse has $h = 22$ electoral districts and $\ell = 26$ rural and urban districts, each with a given population of $w_i \in \mathbb{Z}_{\geq 1}$, $1 \leq i \leq \ell$. Let $G = (V, E)$ be the *contiguity graph* on rural/urban districts V, i.e., $\{i, j\} \in E$ iff areas of $i, j \in V, i \neq j$ have a common border (which is not negligibly short).

APCC Objective In order to minimize population deviations accompanying with apportionment $x \in \text{DivStd}(h, w^{\mathcal{C}})$, define $f(\mathcal{C}, x) := \sum_{1 \leq b \leq p} x_b \cdot \left| \frac{\sum_{i \in C_b} w_i}{x_b} - \frac{w_+}{h} \right|$. Objective f can be formulated linearly with variables of (2): $\min f(\mathcal{C}, x) \iff \min \sum_b z_b$ s.t. $\sum_{i \in V} w_i \delta_{i,b} - \frac{w_+}{h} x_b \leq z_b$, $\frac{w_+}{h} x_b - \sum_{i \in V} w_i \delta_{i,b} \leq z_b$, $z_b \geq 0 \; \forall b$.

Additional Constraints To get feasible PD instances, each coalition needs at least one electoral district, i.e., $x_b \geq \gamma_b \; \forall b$. In order not to dilute administrative conformity too much, add an upper bound $x_b \leq x^{\max} \; \forall b$. We choose $x^{\max} := 4$.

In our implementation, we utilize separator inequalities as lazy constraints to ensure connectedness [4, 14], we add $\gamma_b \geq \gamma_{b+1} \; \forall b \in B \setminus \{\ell\}$ and fix $\gamma_b = 1$ for $b = 1, \ldots, \lceil \frac{h}{x^{\max}} \rceil$ in purpose of symmetry breaking.

Using GUROBI as MILP solver, we perform five independent runs with a timelimit of 10 min each. The best solution found is shown in Fig. 1. Unfortunately, its quality in terms of an optimality gap can not be assessed, since the solver was not able to compute a dual bound better than the trivial one, i.e., 0.

The latter fact leaves room for further research. In addition, a link to a suitable PD algorithm for designing electoral districts in coalitions with $x_b > 1$ should be implemented—perhaps even with a response to DivCoalStd in order to be able to evaluate coalitions there on the basis of designed electoral districts.

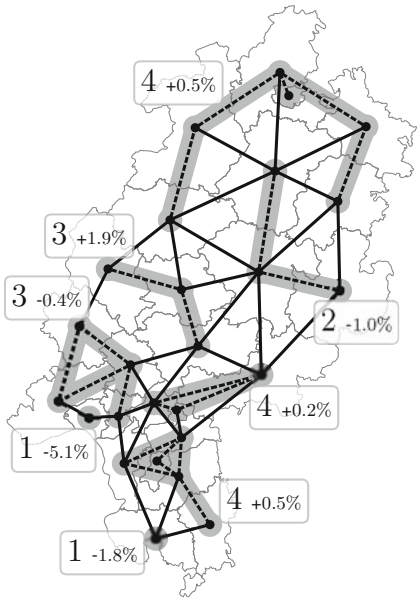


Fig. 1 Contiguity graph of the 26 rural and urban districts of the German federal state of Hesse, and the best found solution of DivCoalStd^f: The solution consists of eight connected coalitions (grey, dashed edges). Next to each coalition, a two-number box (such as 4, +0.5%) indicates the electoral district apportionment of that coalition ($4 = x_b$), and resulting (average) population deviation (+0.5%). Population figures used are of June 2015

References

1. Balinski, M.L., Young, H.P.: Criteria for proportional representation. Oper. Res. **27**(1), 80–95 (1979)
2. Balinski, M.L., Young, H.P.: Fair Representation: Meeting the Ideal of One Man, One Vote. Brookings Institution Press, Washington (1982)
3. Dyer, M.E., Frieze, A.M.: On the complexity of partitioning graphs into connected subgraphs. Discret. Appl. Math. **10**(2), 139–153 (1985)
4. Fischetti, M., Leitner, M., Ljubic, I., Luipersbeck, M., Monaci, M., Resch, M., Salvagnin, D., Sinnl, M.: Thinning out Steiner trees: a node-based model for uniform edge costs. Math. Program. Comput. **9**(2), 203–229 (2016)
5. Goderbauer, S.: Mathematische Optimierung der Wahlkreiseinteilung für die Deutsche Bundestagswahl. Springer, Basel (2016)
6. Goderbauer, S.: Political districting for elections to the German bundestag: an optimization-based multi-stage heuristic respecting administrative boundaries. In: Operations Research Proceedings 2014, pp. 181–187. Springer, Basel (2016)
7. Goderbauer, S., Wicke, M.: Constituencies for German federal elections: legal requirements and their observance, Tech. report. repORt 2017-041, Lehrstuhl für Operations Research, RWTH Aachen University (2017)
8. Goderbauer, S., Winandy, J.: Political districting problem: literature review and discussion with regard to federal elections in Germany (2018, under revision)
9. Johnson, D.S.: The NP-completeness column: an ongoing guide. J. Algorithms **3**, 182–195 (1982)
10. Kopfermann, K.: Mathematische Aspekte der Wahlverfahren: Mandatsverteilung bei Abstimmungen. BI-Wissenschaftsverlag, Mannheim (1991)
11. Pukelsheim, F.: Proportional Representation. Springer, Basel (2017)
12. Ricca, F., Scozzari, A., Simeone, B.: Political districting: from classical models to recent approaches. 4OR **9**(3), 223–254 (2011)

13. Sainte-Laguë, A.: La représentation proportionnelle et la méthode des moindres carrés. Annales scientifiques de l'Ecole normale supérieure **27**, 529–542 (1910)
14. Wang, Y., Buchanan, A., Butenko, S.: On imposing connectivity constraints in integer programs. Math. Program. **166**(1–2), 241–271 (2017)

Beautification of City Models Based on Mixed Integer Linear Programming

Steffen Goebbels and Regina Pohle-Fröhlich

1 Introduction

Most algorithms that generate 3D building models either follow a data-driven or a model-driven methodology or combine both methodologies, see [10]. In a model-driven approach, parameterized standard roofs from a catalogue are fitted to point clouds obtained by airborne laser scanning or photogrammetry. Data-driven algorithms estimate plane segments and combine them to watertight roofs (cf. [5–7, 9]). However, publicly available point clouds often are sparse and contain less than ten points per square meter. This makes it difficult to estimate roof polygons exactly. In our data-driven framework, we use a Ramer Douglas Peucker algorithm to obtain straight edges, see [2, 3]. We also estimate rectangular structures and snap vertices to intersection lines between estimated planes (ridge lines). Nevertheless, the models still need improvement, especially along step edges, see Fig. 2. Such model beautification also is useful in connection with reverse engineering of scanned 3D objects to CAD models, see [1] and the literature cited there.

The idea of this paper is to cautiously change the positions of vertices to obtain a maximum number of edges that are orthogonal to edges of the cadastral footprint. A changed vertex position has impact on at least two edges and the angles between them and their neighbor edges. Thus a global optimization problem arises. A similar problem is addressed in [11] and solved by minimizing a non-linear energy function through graph reduction techniques. However, amongst others, the condition of orthogonality is linear. Linear programming has already been successfully used to planarize roof polygons of city models, see [4]. For model beautification, binary variables are required to select the vertices that have to be changed. Thus, we

S. Goebbels (✉) · R. Pohle-Fröhlich
iPattern Institute, Niederrhein University of Applied Sciences, Krefeld, Germany
e-mail: steffen.goebbels@hs-niederrhein.de; Steffen.Goebbels@hsnr.de; regina.pohle@hs-niederrhein.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_16

introduce a mixed integer linear program (MIP) in the next section. The last section summarizes results.

2 A Linear Program for Beautification

Let D be a set of normalized direction vectors. Each vector of D is parallel or orthogonal to a significant edge of the building's cadastral footprint. We regard an edge as significant if it is longer than 2 m. We only consider the longest cadastral edges and limit D to eight elements. Each vector $d \in D$ is represented as a pair $(d.x, d.y)$, $\sqrt{(d.x)^2 + (d.y)^2} = 1$.

All building edges are projected to the x-y-ground plane. To reduce the number of edges and computational complexity, we first collapse sequences of edges to one edge if they have nearly the same orientation, i.e. angles between edges of the sequence are near π, and if the surrogate edge does not differ more than a threshold value from the replaced edges. The outcome is a planar graph. We determine all connected components of this graph because coordinate changes of a vertex do only have impact on the connected component of the vertex. For example, dormers in the roof's interior might lead to separate connected components.

Let E be the set of all non-trivial 2D edges $e = (e_1, e_2) = ((e_1.x, e_1.y), (e_2.x, e_2.y))$ of a connected component that fulfill following two conditions (see Fig. 1): Each edge has to possess at least one vertex e_1 or e_2 that does not coincide with a vertex of the outer or inner cadastral footprint polygons. Also, an edge must not be completely covered by a footprint edge. We split $E = E' \cup E''$ into two disjoint sets. Edges in E' are allowed to change their orientation, whereas edges in E'' have to keep their original direction. We put all edges into E'' that are originally orthogonal to a vector of D. By keeping their orientation, we will reduce the number of binary variables.

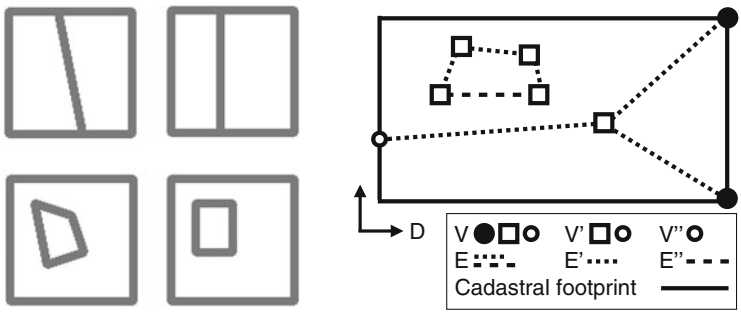


Fig. 1 Left: two examples with a square as building footprint. The left sketches show original layouts, the right ones show the outcomes of optimization. The upper example has an edge with two vertices that are placed on footprint edges. The vertices have to stay on these edges. The vertices of the interior polygon in the lower example can be moved without such restrictions. Right: a roof's graph with two connected components is shown to illustrate the sets of edges $E = E' \cup E''$, the sets of vertices $V'' \subset V' \subset V$, and the set of footprint directions D

V denotes the set of vertices belonging to edges in E, $V' \subset V$ is the set of vertices for which we allow position changes. Vertices in V' must not be used in footprint polygons and they must not represent intersection points of estimated roof's ridge lines. There might be vertices in V' that are not within the interior of the footprint but are positioned on a footprint edge. Let $V'' \subset V'$ be the set of these vertices. Our MIP changes their positions as well as positions of interior vertices. To this end we need float variables x_v^+, x_v^-, and y_v^+, y_v^- for each $v \in V$ that represent non-negative distances. For $v \in V \setminus V'$ their values are fixed set to zero.

We introduce binary variables $x_{e,d}$ which indicate that an edge $e \in E'$ becomes orthogonal to a direction vector $d \in D$:

$$x_{e,d} = \left\{ \begin{array}{l} 1 : e \text{ is transformed to become orthogonal to } d \\ 0 : \text{else} \end{array} \right\} \text{ for } e \in E', d \in D.$$

With a threshold value ε that limits coordinate changes, the MIP optimizes an objective function with the primary goal to find a maximum number of orthogonal edges:

$$\text{maximize} \left(\sum_{e \in E'} \sum_{d \in D} x_{e,d} \right) - \frac{1}{8 \cdot |V'| \cdot \varepsilon} \sum_{v \in V'} \left(x_v^+ + x_v^- + y_v^+ + y_v^- \right).$$

The secondary aim is to minimize coordinate changes. The factor $\frac{1}{8 \cdot |V'| \cdot \varepsilon}$, in which $|V'|$ is the number of elements of V', ensures that all possible coordinate changes contribute significantly less than one binary variable. We maximize the objective function subject to a list of restrictions:

Modified edges $e \in E'$ have to be orthogonal to direction $d \in D$ if $x_{e,d} = 1$:

$$\begin{aligned} -M \cdot (1 - x_{e,d}) \leq (e_2.x + x_{e_2}^+ - x_{e_2}^- - e_1.x - x_{e_1}^+ + x_{e_1}^-) \cdot d.x \\ + (e_2.y + y_{e_2}^+ - y_{e_2}^- - e_1.y - y_{e_1}^+ + y_{e_1}^-) \cdot d.y \leq M \cdot (1 - x_{e,d}), \end{aligned}$$

here M is chosen as a positive number that is larger than the longest occurring edge.

Thus, we implement the two conditions

$$\begin{aligned} & (x_{e_2}^+ - x_{e_2}^- - x_{e_1}^+ + x_{e_1}^-) \cdot d.x + (y_{e_2}^+ - y_{e_2}^- - y_{e_1}^+ + y_{e_1}^-) \cdot d.y + M \cdot x_{e,d} \\ & \leq M + (e_1.x - e_2.x) \cdot d.x + (e_1.y - e_2.y) \cdot d.y, \\ & (-x_{e_2}^+ + x_{e_2}^- + x_{e_1}^+ - x_{e_1}^-) \cdot d.x + (-y_{e_2}^+ + y_{e_2}^- + y_{e_1}^+ - y_{e_1}^-) \cdot d.y + M \cdot x_{e,d} \\ & \leq M + (-e_1.x + e_2.x) \cdot d.x + (-e_1.y + e_2.y) \cdot d.y. \end{aligned}$$

We do not change vertices on the cadastral footprint and intersection points of ridge lines. This gives the next set of restrictions:

$$x_v^+ = x_v^- = y_v^+ = y_v^- = 0 \text{ for all } v \in V \setminus V'.$$

Other vertices can be moved but only within the threshold distance $\varepsilon > 0$:

$$0 \leq x_v^+, x_v^-, y_v^+, y_v^- < \varepsilon \text{ for all } v \in V'.$$

This condition might be changed to an adaptive one based on edge length.

Vertices in V'' are only allowed to change their positions so that they stay on the one footprint edge that they are positioned on, see upper example on the left side of Fig. 1. Thus, for all $v \in V''$ we introduce a parameter variable $0 \leq r_v \leq 1$ and require

$$\begin{aligned} x_v^+ - x_v^- - (b_v.x - a_v.x) \cdot r_v &= a_v.x - v.x \\ y_v^+ - y_v^- - (b_v.y - a_v.y) \cdot r_v &= a_v.y - v.y, \end{aligned}$$

where a_v and b_v denote the 2D vertices of the corresponding footprint edge.

To reduce the degrees of freedom, we set $x_{e,d} = 0$ for all $e \in E'$, $d \in D$ with

$$\frac{|(e_2.x - e_1.x) \cdot d.x + (e_2.y - e_1.y) \cdot d.y|}{\sqrt{(e_2.x - e_1.x)^2 + (e_2.y - e_1.y)^2}} > \left|\cos\left(\frac{\pi}{2} + \alpha\right)\right|$$

such that for each model edge we only consider roughly orthogonal cadastral footprint directions. Maximum deviation from $\pm\frac{\pi}{2}$ is determined by the threshold angle α. Another set of restrictions keep the orientation of edges that are orthogonal to a footprint direction $d \in D$ from the beginning: For all $e \in E''$ let

$$(x_{e_2}^+ - x_{e_2}^- - x_{e_1}^+ + x_{e_1}^-) \cdot (e_1.y - e_2.y) + (y_{e_2}^+ - y_{e_2}^- - y_{e_1}^+ + y_{e_1}^-) \cdot (e_2.x - e_1.x) = 0.$$

Finally, feasible solutions fulfil $\sum_{d \in D} x_{e,d} \leq 1$ for all $e \in E'$.

We follow an optimistic approach and, for performance reasons, do not check if edges cross other edges due to position changes. In that case, self intersections of polygons or intersections between different connected components occur. Our existing framework for building model generation resolves such situations by cutting polygons into pieces. If a problem cannot be resolved then the threshold value ε will be reduced and the model will be re-computed. The algorithm starts with a given threshold value ε and then iteratively divides the threshold value by two. However, we only update vertices with their optimized positions if they do not leave the area of the cadastral footprint.

3 Results

Our implementation is based on the GNU Linear Programming Kit library GLPK [8]. We apply the MIP to city model generation of the square kilometer with 1829 buildings of Krefeld that covers the building of our institute. This tile is visualized in Fig. 2. Table 1 summarizes resulting sizes of non-trivial problem instances with

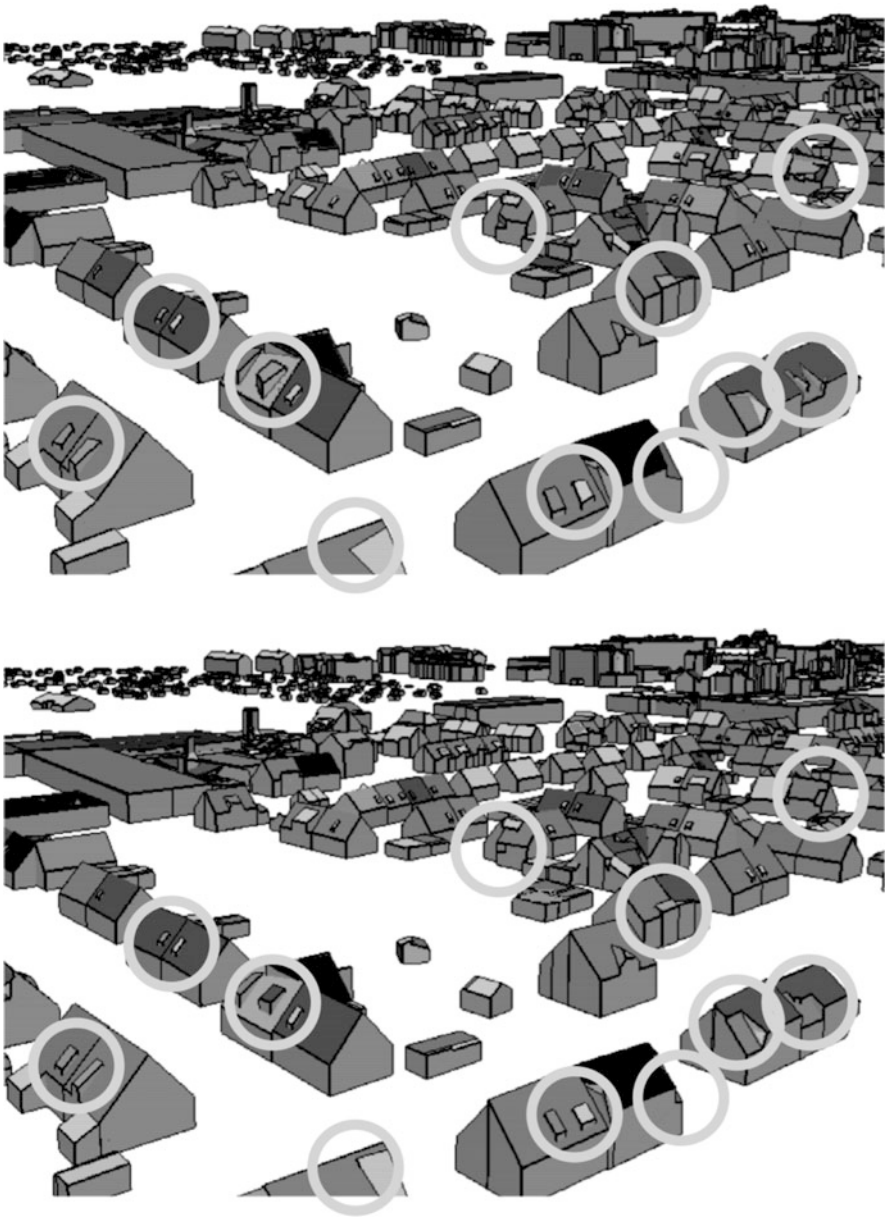


Fig. 2 City model before and after beautification: some changes are marked with circles. Roofs and walls are simplified differently due to post-processing

Table 1 Results for a city model of a square kilometer (UTM intervals $[32{,}330{,}000, 32{,}331{,}000] \times [5{,}687{,}000, 5{,}688{,}000]$: for parameter combination $\varepsilon = 1$ m, $\alpha = \frac{\pi}{6}$, there exist 1826 non-trivial problem instances for connected components, 61 reach the time limit of 2 s (not included in running time data, a larger time limit does not increase the number of successful instances significantly)

		Minimum	Maximum	Arithmetic mean	Median	Quartiles
$\varepsilon = 1\,\text{m}, \alpha = \frac{\pi}{6}$	Variables	9	2709	66.69	35	20, 45
	Binary variables	1	1210	10.34	4	2, 8
	Conditions	2	3004	43.61	20	11, 39
	Changed vertices	0	233	4.01	3	2, 5
	Running time [ms]	0.09	1697.96	17.45	0.46	0.29, 1.28
$\varepsilon = 2\,\text{m}, \alpha = \frac{\pi}{4}$	Variables	9	2951	68.05	36	20, 45
	Binary variables	1	1500	12.51	4	2, 9
	Conditions	2	3679	48.88	22	11, 42
	Changed vertices	0	261	4.44	3	2, 5
	Running time [ms]	0.12	1765.27	19.08	0.52	0.31, 1.51

There are 1876 instances for parameters $\varepsilon = 2$ m, $\alpha = \frac{\pi}{4}$, of which 89 exceed the time limit. Running times are measured on one kernel of a Macbook Pro (2013) with 2.4 GHz i5 processor

at least one changeable edge, i.e. one binary variable. The improved city model is visualized in Fig. 2.

An artifact of optimization with an l_1-norm instead of using least squares is that larger individual coordinate changes might occur. For example, in Fig. 1 the vertical line in the upper left sketch is not optimized to be in the middle of the square.

References

1. Gauthier, S., Puech, W., Bénière, R., Subsol, G.: Orientation beautification of reverse engineered models. In: Proceedings International Conference on Computer Graphics Theory and Applications (GRAPP 2018), pp. 91–100. SCITEPRESS, Funchal (2018)
2. 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)
3. Goebbels, S., Pohle-Fröhlich, R.: Quality enhancement techniques for building models derived from sparse point clouds. In: Proceedings International Conference on Computer Graphics Theory and Applications (GRAPP 2017), pp. 93–104. SCITEPRESS, Porto (2017)
4. Goebbels, S., Pohle-Fröhlich, R., Rethmann, J.: Planarization of CityGML models using a linear program. In: Operations Research Proceedings (OR 2016 Hamburg), pp. 591–597. Springer, Berlin (2016)
5. Haala, N., Kada, M.: An update on automatic 3D building reconstruction. ISPRS J. Photogramm. Remote Sens. **65**, 570–580 (2010)
6. He, Y.: Automated 3D Building Modeling from Airborne LiDAR Data. PhD 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. Makhorin, A.: The GNU Linear Programming Kit (GLPK). Free Software Foundation, Boston (2009)
9. 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)
10. Wang, R., Peethambaran, J., Dong, C.: Lidar point clouds to 3D urban models: a review. IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens. **11**(2), 606–627 (2018)
11. Yan, J., Zhang, K., Zhang, C., Chen, S.C.: Automatic construction of 3-D building model from airborne lidar data through 2-D snake algorithm. IEEE Trans. Geosci. Remote Sens. **53**(1), 3–14 (2015)

Sweep Algorithms for the Capacitated Vehicle Routing Problem with Structured Time Windows

Christoph Hertrich, Philipp Hungerländer, and Christian Truden

1 Introduction

Attended Home Delivery (AHD) services, e.g., online grocery shopping services, have encountered a significant growth in popularity in recent years. In a typical setup, customers choose time windows during which they want to receive their goods from a set of available time windows that is provided by the supplying company. This set is called *structured* if the number of customers is significantly larger than the number of time windows and all windows are pair-wise non-overlapping. Once all orders have been placed, the supplier aims to minimize the fulfillment costs. This involves solving a *capacitated Vehicle Routing Problem with structured Time Windows* (cVRPsTW), which is a special case of the *capacitated Vehicle Routing Problem with Time Windows* (cVRPTW). Heuristics are the method of choice in practice to produce high-quality solutions in reasonable time since the problem instances occurring are typically rather large.

So-called *cluster-first, route-second* methods have been proven to be effective for the classical *capacitated Vehicle Routing Problem* (cVRP). These methods first partition the customers into subsets that are small enough such that they can all be visited by one vehicle. In a second step, they compute a route for each vehicle. The most prominent example of this group of algorithms is the *sweep algorithm* [3]. It clusters the customers by dividing the plane into radial sectors originating from the depot's location. In this work we first generalize the sweep algorithm to the cVRPTW. Secondly, we propose two variants that exploit the additional structure

C. Hertrich (✉)
Department of Mathematics, Technische Universität Kaiserslautern, Kaiserslautern, Germany
e-mail: hertrich@math.tu-berlin.de

P. Hungerländer · C. Truden
Department of Mathematics, Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
e-mail: philipp.hungerlaender@aau.at; christian.truden@aau.at

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_17

of the time windows in a cVRPsTW. Due to the imposed structure the total number of time windows is quite small, which allows us to use window-dependent angles. After obtaining a first feasible solution, improvement heuristics are applied that try to decrease the objective function by slightly altering the angles obtained by our sweep algorithms.

Finally, we conduct an extensive computational study using a large variety of carefully constructed benchmark instances which show that our approach is capable of finding good initial solutions for instances containing up to 2000 customers within a few seconds. Further, we demonstrate that the proposed improvement heuristics allow us to significantly improve the solution quality within a few minutes. We notice that the performance of the different variants is dependent on the characteristics of the considered instance, e.g., whether vehicle capacities or time windows are the stronger restriction.

For an overview of exact resp. heuristic methods for the cVRP(TW) we refer to [1, 2]. Solomon [6] proposes a sweep heuristic that takes time window constraints into account. In contrast to our approach, it does not consider the time windows of the customers when partitioning them. Instead, the time windows are only respected when computing the routes for each vehicle. While Solomon's approach utilizes an insertion heuristic to obtain the routes, we apply a Mixed-Integer Linear Program (MILP) to decide the feasibility of an assignment of customers to a vehicle, as well as to obtain the optimal solution of the occurring routing subproblems.

2 Formal Problem Definition

A cVRPsTW instance consists of a set of time windows $\mathcal{W} = \{w_1, \dots, w_q\}$, where each $w_i \in \mathcal{W}$ is defined through its start time s_{w_i} and its end time e_{w_i} with $s_{w_i} < e_{w_i}$, a set of customers $\mathcal{C}$, $|\mathcal{C}| = n$, a time window assignment function $w\colon \mathcal{C} \to \mathcal{W}$, a depot d from which all vehicles depart from and return to, $\overline{\mathcal{C}} := \mathcal{C} \cup \{d\}$, a travel time function $t\colon \overline{\mathcal{C}} \times \overline{\mathcal{C}} \to \mathbb{R}_{\geq 0}$, a service time function $s\colon \mathcal{C} \to \mathbb{R}_{>0}$, a common vehicle capacity $C \in \mathbb{R}_{>0}$, and an order weight function $c\colon \mathcal{C} \to]0, C]$. We require the windows to be structured, i.e., $n \gg q$ and for $1 \leq i < j \leq q$ it holds $s_{w_j} \geq e_{w_i}$. Moreover, each element in $\overline{\mathcal{C}}$ has coordinates in the two-dimensional plane. We assume that the travel times are correlated to the geographical distances, but not purely determined by them.

A *tour* consists of a set $\mathcal{A} = \{a_1, a_2, \dots, a_k\}$ of customers with corresponding arrival times $\alpha_{a_1}, \dots, \alpha_{a_k}$ during which the vehicles are scheduled to arrive. A tour is called *capacity-feasible*, if $\sum_{i=1}^{k} c(a_i) \leq C$. Moreover, we call it *time-feasible*, if every customer is served within its assigned time window, i.e., $s_{w(a_i)} \leq \alpha_{a_i} \leq e_{w(a_i)}$, $i = 1, \dots, k$, and if there is sufficient time to respect the required service and travel times, i.e., $\alpha_{a_{i+1}} - \alpha_{a_i} \geq s(a_i) + t(a_i, a_{i+1})$, $i = 1, \dots, k-1$. A *schedule* $\mathcal{S} = \{\mathcal{A}, \mathcal{B}, \dots\}$ is a set of tours where each customer occurs in exactly one tour. It is called *feasible*, if all tours are capacity- and time-feasible.

We consider three objectives: The first one is the *number of vehicles used*, i.e. $\lambda_1(\mathcal{S}) := |\mathcal{S}|$. Secondly, the *schedule duration* $\lambda_2(\mathcal{S})$ is defined as the sum of all *tour durations*, i.e. $\lambda_2(\mathcal{A}) := t(d, a_1) + \alpha_{a_k} - \alpha_{a_1} + s(a_k) + t(a_k, d)$. Thirdly, the *schedule travel time* $\lambda_3(\mathcal{S})$ is defined as the sum of all *tour travel times*, i.e. $\lambda_3(\mathcal{A}) := t(d, a_1) + \sum_{i=1}^{k-1} t(a_i, a_{i+1}) + t(a_k, d)$. Similar to Solomon [6], we aim to minimize these three objectives with respect to the lexicographical order $(\lambda_1, \lambda_2, \lambda_3)$, since providing a vehicle is usually the most expensive cost component, followed by the drivers' salaries, and the costs for fuel.

3 Sweep Algorithms for Structured Time Windows

In this section we describe several variants of the sweep algorithm. First, we introduce some more notation and definitions.

Tree-Feasibility For a tour $\mathcal{A}$ let $\mathcal{A}_w \subseteq \mathcal{A}$ be the set of customers assigned to $w \in \mathcal{W}$. We consider the complete directed graph with vertex set $\mathcal{A}_w$ and edge weights $t(a_i, a_j) + s(a_i)$ assigned to each edge (a_i, a_j). The existence of a spanning arborescence through $\mathcal{A}_w$ with length of at most $e_w - s_w$ for all time windows w forms a necessary condition for the time-feasibility of a tour $\mathcal{A}$. In this case we call $\mathcal{A}$ *tree-feasible*. Existence of such an arborescence can be checked in $\mathcal{O}(|\mathcal{A}_w|^2)$ time [7]. Hence, time-infeasibility of a potential tour can often be detected without solving a time-consuming MILP. Moreover, we apply the concept of tree-feasibility in the Corrective Sweep algorithm below.

Angles of Customers Any sweep algorithm is based on the polar coordinate representation of the customers $\mathcal{C}$, where the depot d forms the origin of the coordinate system and $\theta(a) \in [0, 2\pi[$ denotes the angle component of a customer $a \in \mathcal{C}$. The direction of the zero angle θ_0 is a choice parameter of the algorithm as it impacts the result. We choose the zero angle such that it separates the two consecutive customers with the largest angle gap, i.e., $\max_{a,b \in \mathcal{C}} \theta(a) - \theta(b)$ is minimized. Moreover, we run all algorithms in clockwise and counterclockwise direction and in each case select the variant that gives the better result.

In the following, we denote $\mathcal{C}(\theta, \theta') = \{a \in \mathcal{C} \mid \theta(a) \in [\theta, \theta'[\}$, $\mathcal{C}_w = \{a \in \mathcal{C} \mid w(a) = w\}$, and $\mathcal{C}_w(\theta, \theta') = \mathcal{C}(\theta, \theta') \cap \mathcal{C}_w$.

3.1 Sweep Strategies

We propose the following general strategy to obtain high-quality solutions for a given cVRPTW or cVRPsTW instance. It consists of three steps:

1. Use a variant of the sweep algorithm to determine a feasible clustering of $\mathcal{C}$.
2. Apply local improvement heuristics to enhance the quality of the clustering.
3. Compute the optimal route for each cluster.

In all three steps, the *Traveling Salesperson Problem with Time Windows* (TSPTW) respectively the *Traveling Salesperson Problem with structured Time Windows* (TSPsTW) occurs as a subproblem to check the time-feasibility or to obtain the optimal solution of a single tour. We apply two MILP formulations that have been proposed in previous work [4], a general one for the TSPTW, and a more efficient one that is tailored to the TSPsTW. Following the lexicographical order, it first minimizes λ_2, and then λ_3 while keeping λ_2 fixed. Next let us relate our notation to the well-known sweep algorithm [3] for the cVRP.

Traditional Sweep The clustering method proposed in [3] relates to Step 1. It finds angles θ_i such that the i-th cluster is given by $\mathcal{C}(\theta_{i-1}, \theta_i)$ as follows: Set $\theta_0 = 0$. For $i = 1, 2, \ldots$ make θ_i as large as possible such that $\mathcal{C}(\theta_{i-1}, \theta_i)$ forms a capacity-feasible cluster. Here, and in all following algorithms, the range for i is chosen such that all customers are scheduled.

Next, we propose a natural generalization of the Traditional Sweep algorithm that works for instances having structured (cVRPsTW), as well as for instances having arbitrary time windows (cVRPTW).

Simple Sweep Choose the angle θ_i as large as possible while ensuring that the resulting cluster $\mathcal{C}(\theta_{i-1}, \theta_i)$ is still small enough such that a time- and capacity-feasible tour that visits all contained customers can be found. We check the time-feasibility using the TSPTW- resp. TSPsTW-MILP.

Now we present two variants of the sweep algorithm that exploit the additional structure of the time windows of cVRPsTW instances.

Window-Wise Sweep In case of structured time windows there are quite few time windows in comparison to customers, i.e., $q \ll n$. Hence, we can define window-dependent angles θ_i^j, $i \geq 1$, $j = 1, \ldots, q$, such that the i-th cluster is given by $\bigcup_{j=1}^{q} \mathcal{C}_{w_j}(\theta_{i-1}^j, \theta_i^j)$. We propose to add the customers to the clusters window by window. The resulting algorithm is described as follows: While there are unclustered customers in $\mathcal{C}_{w_j}$, make θ_i^j as large as possible such that $\mathcal{C}_{w_j}(\theta_{i-1}^j, \theta_i^j)$ can be added to the i-th cluster while ensuring that the cluster stays time- and capacity-feasible. If necessary, increase the number of clusters.

Corrective Sweep If vehicle capacity is a stronger restriction than time windows, then Window-wise Sweep creates large sectors for the first few time windows and runs out of capacity later on, causing an increased need of vehicles. Therefore, we propose another variant that prevents this from happening. We first initialize angels θ_i^0 of maximal size such that all sets $\mathcal{C}(\theta_{i-1}^0, \theta_i^0)$ are capacity- and tree-feasible. Then we start with empty clusters. For $j = 1, \ldots, q$ we set $\theta_i^j = \theta_i^{j-1}$ and add the customers of $\mathcal{C}_{w_j}$ accordingly. Since this may result in some infeasible clusters, the angles θ_i^j have to be adjusted. Starting with $i = 1$, we check whether the i-th cluster is still feasible. If not, we try to reduce the size of cluster i by increasing the angles $\theta_{i-1}^j, \ldots, \theta_1^j$, such that cluster i becomes feasible while the clusters $1, \ldots, i-1$ remain feasible. If this procedure does not succeed, we decrease θ_i^j until the i-

th cluster becomes feasible. Then, we increment i and repeat this procedure. If necessary, we increase the number of clusters.

Local Improvement Heuristic After obtaining an initial clustering using one of the heuristics described above, we aim to improve the clustering during Step 2. In case that the clustering was obtained using the Simple Sweep algorithm, we set $\theta_i^j = \theta_i$. For each angle θ_i^j we try to improve the objective function iteratively by decreasing or increasing the angle. In each step the angle is slightly altered such that one customer moves to another cluster. A change is accepted if the lexicographical objective is improved. This procedure is repeated until a local minimum is reached.

Routing In Step 3 a tour for each cluster is obtained by solving the TSPTW- or TSPsTW-MILP with lexicographical objective (λ_2, λ_3) to optimality.

4 Computational Experiments

As to our best knowledge, none of the available cVRPTW benchmark instances comply with the considered AHD use-case, we created a new benchmark set that resembles urban settlement structures in order to provide meaningful computational experiments. We placed customers on a 20 km $\times$ 20 km square grid (roughly the size of Vienna). In order to achieve varying customer densities, only 20% of the customer locations are sampled from a two-dimensional uniform distribution. The remaining 80% of the customer locations have been sampled from clusters whose centers and shapes have been randomly sampled. Travel times are calculated proportional to the Euclidean distances rounded to integer seconds. As proposed by Pan et al. [5] we assume a travel speed of 20 km/h. Each customer is randomly assigned to one time window out of a set of ten consecutive time windows, where each is 1 h long. We assume the service time at each customer to be 5 min. The order weights are sampled from a truncated normal distribution centered around 5 units. We consider configurations with $n = 250, 500, 1000, 2000$ and $C = 200, 400$. Choosing $C = 200$ results in the vehicle capacity being the bottleneck, while choosing $C = 400$ causes that the time window constraints are the most limiting factor. For each configuration we constructed 100 instances and report the average results. All instances can be downloaded from http://tinyurl.com/vrpstw.

All experiments were performed on an Ubuntu 14.04 machine equipped with an Intel Xeon E5-2630V3 @ 2.4 GHz 8 core processor and 132 GB RAM. We use Gurobi 8.0 in single thread mode to solve the MILPs. To demonstrate the effectiveness of the improvement heuristics, we compare the case where all three steps are performed against the case where Step 2 is omitted.

In Table 1 we present the results of the computational study for $n = 2000$. In the case that the vehicle capacity is the more limiting factor, i.e., $C = 200$, Simple Sweep and Corrective Sweep produced nearly identical results with respect to all three objectives. In terms of λ_1 both algorithms clearly outperform Window-wise

Table 1 Results for instances with $n = 2000$

Vehicle capacity	200				400			
Runtime/objectives	t	λ_1	λ_2	λ_3	t	λ_1	λ_2	λ_3
Without improvement								
Simple sweep	7.2	54.0	507.7	332.7	41.0	42.1	406.5	315.3
Window-wise sweep	96.9	64.4	381.2	371.5	306.7	35.0	340.2	328.1
Corrective sweep	7.6	54.0	507.2	332.5	34.6	41.5	402.1	314.8
With improvement								
Simple sweep	241.0	54.0	480.3	318.1	1324.7	41.9	368.5	300.8
Window-wise sweep	141.9	64.4	378.5	369.3	679.1	35.0	336.2	324.8
Corrective sweep	248.6	53.9	479.9	318.0	1353.5	41.4	366.8	300.5

We report average values over 100 instances each. The runtime in seconds is denoted by t, while λ_1 denotes the number of vehicles used and λ_2 resp. λ_3 denote the tour duration resp. the travel time in hours

Sweep. However, with respect to λ_2, Window-wise Sweep produced the best results. In the case that time window constraints pose the strongest restriction, i.e., $C = 400$, Window-wise Sweep clearly produced the best results with respect to λ_1 and λ_2. In general, we suggest to apply Corrective Sweep as an algorithm producing good solutions in most cases. If the time window constraints are the most limiting factor of the considered instances, or if λ_2 is more relevant than λ_1, then Window-wise Sweep is the best choice.

As it is rather hard to remove a whole vehicle from a schedule, we notice that applying our local improvement heuristic rarely results in a reduction of the primary objective λ_1. However, the study shows that the improvement heuristic heavily impacts the schedule duration λ_2, while also having positive impact on the travel time λ_3. The experiments show that our heuristics, when applied to large instances containing 2000 customers, take a few seconds to around five minutes to produce a feasible schedule, and below 25 min to produce an improved schedule.

Acknowledgements Christoph Hertrich is supported by the Karl Popper Kolleg "Modeling-Simulation-Optimization" funded by the Alpen-Adria-Universität Klagenfurt and the Carinthian Economic Promotion Fund (KWF).

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. Bräysy, O., Gendreau, M.: Vehicle routing problem with time windows, part I: route construction and local search algorithms. Transp. Sci. **39**(1), 104–118 (2005)
3. Gillett, B.E., Miller, L.R.: A heuristic algorithm for the vehicle-dispatch problem. Oper. Res. **22**(2), 340–349 (1974)

4. Hungerländer, P., Truden, C.: Efficient and easy-to-implement mixed-integer linear programs for the traveling salesperson problem with time windows. Transp. Res. Procedia **30**, 157–166 (2018)
5. Pan, S., Giannikas, V., Han, Y., Grover-Silva, E., Qiao, B.: Using customer-related data to enhance e-grocery home delivery. Ind. Manag. Data Syst. **117**(9), 1917–1933 (2017)
6. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. **35**(2), 254–265 (1987)
7. Tarjan, R.E.: Finding optimum branchings. Networks **7**(1), 25–35 (1977)

An Improved Arcflow Model for the Skiving Stock Problem

John Martinovic, Maxence Delorme, Manuel Iori, and Guntram Scheithauer

1 Introduction

We consider a threshold length $L \in \mathbb{N}$, and $m \in \mathbb{N}$ item types that are specified by their length l_i and frequency (of occurrence) b_i ($i \in I := \{1, \ldots, m\}$). Then, the *one-dimensional skiving stock problem (SSP)* requires to recompose the given items in order to obtain a maximal number of objects each having a length at least L. Originating from applications in paper recycling [7], such objectives are of high interest in industrial production [2, 14] and wireless communications [9] as well. The problem under consideration was introduced in [1] for the special case of highly heterogeneous input lengths and very small quantities b_i ($i \in I$), respectively, and originally termed as the *dual bin packing problem*. Its current name goes back to Zak [14], who presented a first pattern-based modelling approach for arbitrary frequencies. Although this model is conjectured to possess a very tight LP relaxation, similar to the cutting context [6, 10, 12, 13], the number of variables is exponential with respect to the number of items. Consequently, pseudo-polynomial models for the SSP have been established in literature [8] and were shown to exhibit an equally good relaxation. For a long time, research

J. Martinovic (✉) · G. Scheithauer
Institut für Numerische Mathematik, Technische Universität Dresden, Dresden, Germany
e-mail: john.martinovic@tu-dresden.de; guntram.scheithauer@tu-dresden.de

M. Delorme
School of Mathematics, The University of Edinburgh, Edinburgh, UK
e-mail: maxence.delorme@ed.ac.uk

M. Iori
DISMI, Università di Modena e Reggio Emilia, Reggio Emilia, Italy
e-mail: manuel.iori@unimore.it

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_18

on these alternative formulations for discrete optimization problems was rather theoretically motivated; but especially after the publication of the famous book [11] by Nemhauser and Wolsey, also computational aspects (e.g., the strength of the considered models) became a central concern in IP modeling. In recent years, the sharp development of (commercial) MILP software and the rapid progress in terms of powerful hardware components have successively increased the scientific importance of pseudo-polynomial formulations for the solution of cutting and packing problems.

In this article, we introduce a new graph-theoretical ILP formulation for the SSP that is based on the idea of reflected arcs originally presented in [5]. The main novelty of this approach is to only consider half of the bin capacity (i.e., a significantly reduced set of vertices), so that any pattern is decomposed into two subpaths being connected by a reflected arc. Thereby, the number of arcs of the arcflow graph can be decreased considerably compared to the original formulation [8].

In the next section, we briefly review the standard arcflow approach for the SSP. Afterwards, the improved formulation and some related theoretical properties are presented. Then, the computational performance of both models is compared based on randomly generated instances. Finally, we summarize the main ideas of this paper and give an outlook on future research.

2 The Arcflow Model

Let $E = (m, l, L, b)$ with $l = (l_1, \ldots, l_m)^\top \in \mathbb{Z}_+^m$ and $b = (b_1, \ldots, b_m)^\top \in \mathbb{Z}_+^m$ denote an *instance* of the SSP. Without loss of generality, we may assume that $L > l_1 > l_2 > \ldots > l_m \geq 1$ is satisfied. Any combination of items is represented by a vector $a = (a_1, \ldots, a_m)^\top \in \mathbb{Z}_+^m$ (with $a_i \in \mathbb{Z}_+$ counting the items of type $i \in I$), and is referred to as a *pattern* whenever $l^\top a \geq L$ holds. Note that considering the set $P^\star(E)$ of *minimal patterns* (where each appearing item is indeed necessary to ensure $l^\top a \geq L$) is sufficient. Let $\bar{v} := \max\left\{l^\top a \,\middle|\, a \in P^\star(E)\right\}$ describe the maximal length of a minimal pattern. Then, the graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ with $\mathscr{E} = \{(p, q) \in \mathscr{V} \times \mathscr{V} \mid p < L,\ q - p \in \{l_1, \ldots, l_m\}\}$ and $\mathscr{V} = \{0, 1, \ldots, \bar{v}\}$ can be used to model the SSP. Here, an arc $(p, q) \in \mathscr{E}$ indicates the positioning of an item of length $q - p = l_j \in \{l_1, \ldots, l_m\}$ at vertex $p \in \mathscr{V}$. For this approach, different improvements have been discussed in the literature—most importantly the theory of *normal patterns* or *raster points* [3, 4] (restricting $\mathscr{V}$ to only feasible combinations of the item lengths), and symmetry reductions by considering monotonically decreasing paths [8]—leading to a simpler graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$, in

general. Let x_{pq} denote the flow along the arc $(p, q) \in \mathscr{E}$; then the arcflow model [8] is given by:

Arcflow model of the SSP

$$z^{AF} = \max \sum_{(0,q)\in A^+(0)} x_{0q}$$

$$\text{s.t.} \quad \sum_{(p,q)\in A^-(q)} x_{pq} = \sum_{(q,r)\in A^+(q)} x_{qr}, \qquad q \in \mathscr{V}, 0 < q < L, \tag{1}$$

$$\sum_{(p,q)\in E(i)} x_{pq} \le b_i, \qquad i \in I, \tag{2}$$

$$x_{pq} \in \mathbb{Z}_+, \qquad (p, q) \in \mathscr{E}. \tag{3}$$

Note that $A^-(q)$ and $A^+(q)$ model the incoming and emanating arcs of vertex $q \in \mathscr{V}$, respectively, whereas $E(i)$ collects all arcs referring to an item of length l_i. Constraints (1) can be interpreted as a flow conservation, whereas conditions (2) ensure that the given item limitations are respected. Thus, the objective function maximizes the total flow within $\mathscr{G}$, that is, the total number of objects obtained.

3 The Reflect Arcflow Model

As explained in [5], the key idea of the improved approach is to start from the arcflow model, but to represent a bin by using half of its capacity twice. More specifically, each arc whose head would lie in the second half is reflected into the first half. Since skiving stock patterns can exhibit any length $L_t \in \mathscr{L} := \left\{ l^\top a \,\middle|\, a \in P^\star(E) \right\}$ (which are identical to the sinks of $\mathscr{G}$), we usually would have to cope with multiple reflection points $L_t/2$. Moreover, the elements of $\mathscr{L}$ are not known in advance and, hence, cannot be used to efficiently implement a reflect graph. Due to these difficulties, we introduce additional loss arcs in order to force any path (or pattern) to have exactly the length L. Then, [5, Algorithm 7] (with some very minor adaptations to the new context) can be applied to build the reflect graph $\mathscr{H} := (\mathscr{U}, \mathscr{A})$ (for the SSP) with only one reflection point $r := L/2$. Indeed, we have that any arc $(u, v) \in \mathscr{E}$ (appearing in Sect. 2) with either $u < v \le r$ is maintained as a *standard arc* $(u, v, s) \in \mathscr{A}$, or with $u < r < v$ is transformed into a *reflected arc* $(u, L - v, r) \in \mathscr{A}$, or with $r \le u < v$ is not considered anymore. Finally, we add the *loss arcs* (d, e, l) (to connect all nodes d, e, where e is the direct successor of $d \ne 0$) as well as the reflected arc $(L/2, L/2, r)$, see Fig. 1 for an example. Observe that the effect of raster points (as mentioned in Sect. 2) is noticed almost only at the beginning of the graph, whereas it tends to be irrelevant for the second half of the vertices. Hence, the introduced restriction to the node set also significantly reduces the number of arcs compared to the original arcflow formulation.

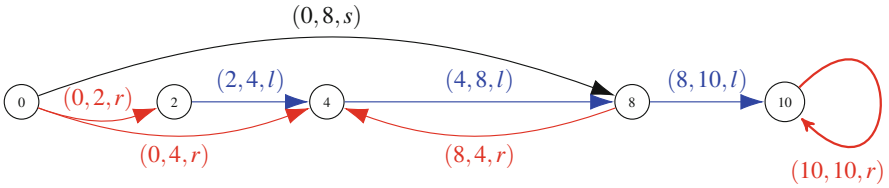


Fig. 1 The reflect graph for the instance $E = (3, (18, 16, 8), 20, (10, 10, 10))$. An optimal solution with $z^{RE} = 15$ is given by $\xi_{0,2,r} = \xi_{0,4,r} = \xi_{0,8,s} = \xi_{2,4,l} = \xi_{8,10,l} = 10$, $\xi_{4,8,l} = 20$, and $\xi_{10,10,r} = -5$. The pattern $a^1 = (0, 2, 0)^\top$ is used 5 times, and can be recomposed by two paths of type $(0, 4, r) \to (4, 8, l) \to (8, 10, l)$ that are connected by the reflected arc $(10, 10, r)$. The pattern $a^2 = (1, 0, 1)^\top$ is used 10 times and consists of the subpaths $(0, 2, r) \to (2, 4, l) \to (4, 8, l)$ and $(0, 8, s)$

Let us define $\mathscr{U}^\star := \mathscr{U} \setminus \{0\}$ and $\mathscr{A}^\star := \mathscr{A} \setminus \{(L/2, L/2, r)\}$, and let $\xi_{de\kappa}$ denote the flow along the generic arc $(d, e, \kappa) \in \mathscr{A}$ (where $\kappa \in \{s, r, l\}$ is possible); then the reflect arcflow model is given by

Reflect Arcflow Model of the SSP

$$z^{RE} = \max \sum_{(d,e,r)\in\mathscr{A}} \xi_{der}$$

$$\text{s.t.} \sum_{(d,e,s)\in\mathscr{A}:\, e-d=l_i} \xi_{des} + \sum_{(d,e,r)\in\mathscr{A}:\, e=L-d-l_i} \xi_{der} \le b_i, \qquad i \in I, \tag{4}$$

$$\sum_{(0,e,s)\in\mathscr{A}} \xi_{0es} + \sum_{(0,e,r)\in\mathscr{A}} \xi_{0er} = 2 \sum_{(d,e,r)\in\mathscr{A}} \xi_{der}, \tag{5}$$

$$\sum_{(d,e,s)\in\mathscr{A}} \xi_{des} + \sum_{(e,f,l)\in\mathscr{A}} \xi_{efl} = \ldots$$

$$\ldots \sum_{(d,e,r)\in\mathscr{A}} \xi_{der} + \sum_{(d,e,l)\in\mathscr{A}} \xi_{del} + \sum_{\kappa\in\{r,s\}} \sum_{(e,f,\kappa)\in\mathscr{A}} \xi_{ef\kappa}, \qquad e \in \mathscr{U}^\star, \tag{6}$$

$$\sum_{(d,e,l)\in\mathscr{A}} \xi_{del} + \sum_{(d,e,r)\in\mathscr{A}} \xi_{der} \ge \sum_{(e,f,l)\in\mathscr{A}} \xi_{efl}, \qquad e \in \mathscr{U}^\star, \tag{7}$$

$$\xi_{de\kappa} \in \mathbb{Z}_+, \qquad (d, e, \kappa) \in \mathscr{A}^\star, \tag{8}$$

$$\xi_{L/2,L/2,r} \in \mathbb{Z}. \tag{9}$$

Since each unit of flow on a reflected arc $(d, e, r) \in \mathscr{A}$ refers to one single pattern, the objective function maximizes the number of constructed patterns. Moreover, conditions (4) require each item to be used at most b_i times, whereas Eq. (5) basically says that two subpaths (starting at $v = 0$) are needed per pattern (i.e., per unit of flow on a reflected arc). Constraints (6) model the flow conservation at interior vertices: each flow entering a node through a standard arc has to be continued by a flow leaving the node through a standard or a reflected arc (classical flow conservation), or by a flow entering the node through a reflected arc (connection between two subpaths), including the additional flow brought and removed by the loss arcs. Note that a loss arc may only follow another loss arc or a reflected arc, see (7). One particularity of this model is that we have to allow negative flows on the reflected arc $(L/2, L/2, r)$ to enable two reflected paths to be merged together. This is required when large items have to be packed together in the optimal solution, see Fig. 1. Altogether, modeling and generating the reflect arcflow formulation for the SSP is more challenging than in the cutting scenario and requires (partly) independent techniques.

4 Computational Results

In order to evaluate the computational behavior of both models, we randomly generated 20 instances each for different pairs (m, L) of input data (specified in Table 1) and collected the following averaged quantities: numbers of variables n_{var} and constraints n_{con}, solution time t (in s). The item lengths and availabilities were chosen from uniformly distributed integers $l_i \in [L/10, 3/4 \cdot L]$ and $b_i \in [1, 100]$, respectively. All our experiments were executed on a Quad-Core AMD A10-5800K with 3.8 GHz and 16 GB RAM, using Gurobi 7.5.2 as ILP solver.

Based on the data from Table 1, it can clearly be seen that the original arcflow model is outperformed by the reflect formulation. Indeed, both models solved to

Table 1 Averaged computational results for different choices of (m, L)

		$m = 50$		$m = 100$		$m = 200$	
		Arcflow	Reflect	Arcflow	Reflect	Arcflow	Reflect
$L = 1000$	t	0.43	0.08	2.11	0.32	10.07	2.20
	n_{var}	8689.95	1473.30	23503.40	4729.10	62652.10	13253.75
	n_{con}	711.75	634.20	871.80	954.90	1021.60	1163.80
$L = 2500$	t	0.92	0.12	7.86	0.55	36.07	5.05
	n_{var}	14265.60	1839.10	47986.25	6822.55	126413.55	20922.00
	n_{con}	1486.00	1202.00	1866.90	1883.60	2135.90	2470.70
$L = 5000$	t	1.86	0.15	20.41	1.02	101.62	9.13
	n_{var}	20405.10	2080.25	74619.15	8377.60	210507.55	28205.00
	n_{con}	2496.05	1621.55	3331.90	3094.00	3850.25	4290.10

proven optimality all attempted instances, but reflect has been much quicker. More precisely, for any of the 180 tested instances, the new approach led to significant savings in terms of variables (ranging from 77% to 92%, with an average of 85%) and solution times (ranging from 50% to 97%, with an average of 87%). As regards the number of constraints, there is no clear relationship between the corresponding quantities. On the one hand, our new approach allows for using much fewer flow conservation constraints (in most cases only roughly 50% of the original number), whereas, on the other hand, an additional set of constraints (see (7)) has to be included.

5 Conclusions

In this paper we investigated an improved arcflow formulation for the skiving stock problem. The main idea of this approach is to only consider the first half of the vertex set, and to model a pattern as the sum of two subpaths that are connected by a reflected arc. As we have shown in the computational experiments, the new model possesses significantly fewer variables and much better solution times compared to the original formulation. Consequently, the reflect arcflow model may be seen as a powerful tool for solving (large) instances of the SSP in reasonably short time.

References

1. Assmann, S.F., Johnson, D.S., Kleitman, D.J., Leung, J.Y.-T.: On a dual version of the one-dimensional Bin Packing Problem. J. Algorithms **5**, 502–525 (1984)
2. Chen, Y., Song, X., Ouelhadj, D., Cui, Y.: A heuristic for the skiving and cutting stock problem in paper and plastic film industries. Int. Trans. Oper. Res. **26**(1), 157–179 (2019)
3. Côté, J.-F., Iori, M.: The meet-in-the-middle principle for cutting and packing problems. INFORMS J. Comput. **30**(4), 646–661 (2018)
4. Christofides, N., Whitlock, C.: An algorithm for two-dimensional cutting problems. Oper. Res. **25**(1), 30–44 (1977)
5. Delorme, M., Iori, M.: Enhanced pseudo-polynomial formulations for bin packing and cutting stock problems. Technical Report OR-17-6, DEI "Guglielmo Marconi", Università di Bologna (2017)
6. Delorme, M., Iori, M., Martello, S.: Bin packing and cutting stock problems: mathematical models and exact algorithms. Eur. J. Oper. Res. **255**, 1–20 (2016)
7. Johnson, M.P., Rennick, C., Zak, E.J.: Skiving addition to the cutting stock problem in the paper industry. SIAM Rev. **39**(3), 472–483 (1997)
8. Martinovic, J., Scheithauer, G.: Integer linear programming models for the skiving stock problem. Eur. J. Oper. Res. **251**(2), 356–368 (2016)
9. Martinovic, J., Jorswieck, E., Scheithauer, G., Fischer, A.: Integer linear programming formulations for cognitive radio resource allocation. IEEE Wirel. Commun. Lett. **6**(4), 494–497 (2017)
10. Martinovic, J., Scheithauer, G., Valério de Carvalho, J.M.: A comparative study of the arcflow model and the one-cut model for one-dimensional cutting stock problems. Eur. J. Oper. Res. **266**(2), 458–471 (2018)

11. Nemhauser, G., Wolsey, L.: Integer and Combinatorial Optimization. Wiley, New York (1988)
12. Scheithauer, G.: Introduction to Cutting and Packing Optimization – Problems, Modeling Approaches, Solution Methods. Springer, Chur (2018)
13. Valério de Carvalho, J.M.: LP models for bin packing and cutting stock problems. Eur. J. Oper. Res. **141**(2), 253–273 (2002)
14. Zak, E.J.: The skiving stock problem as a counterpart of the cutting stock problem. Int. Trans. Oper. Res. **10**, 637–650 (2003)

Simulation-Based Location Optimization of Ambulance Stations

Johanna Schneider and Michael Schröder

1 Introduction

Emergency medical services (EMS) are an important part of the public health care system. In case of an emergency, the chance of survival is dependent on the time until an ambulance arrives. Therefore, the response time to emergency calls is an important criteria to evaluate the location structure of ambulance stations.

The response time mainly consists of the driving time from the ambulance's current position to the rescue location. In this paper, we outline how to estimate the driving time for urgent rescue operations (blue lights and siren are used) based on real rescue data and a detailed road network. Further, we present both a static and dynamic analysis of the location structure of ambulances. For the static analysis, assuming all ambulances are available at their station, we use so-called isochrones. By overlaying these, we calculate the estimated driving time to every road segment from any ambulance station around. This analysis is important for the fulfillment of legal regulations. In order to capture the dynamics, we implemented a discrete event simulation where ambulances can be alarmed also when driving empty. This dynamic analysis gives a realistic estimation of driving times and the expected workload of ambulances. The methods presented in this paper were used for a real world study conducted for a German county council. In a current project HealthFaCT, funded by the German Federal Ministry of Education and Research, we apply, improve and expand these techniques.

In the literature, EMS have been studied a lot, see [1] for a review. Discrete event simulations are presented in [2] and [3]. In both simulations, the area was divided into small subareas with a given travel time in between. This method is applicable

J. Schneider (✉) · M. Schröder
Fraunhofer Institute for Industrial Mathematics ITWM, Kaiserslautern, Germany
e-mail: johanna.schneider@itwm.fraunhofer.de;
https://www.itwm.fraunhofer.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_19

for urban areas but not sufficient in our case. In [4] a discrete event simulation on a detailed road network is presented, but as in [2] and [3] the ambulances can not be alarmed on the road which is possible in our simulation, an important feature for getting realistic estimates.

2 Methods

The crucial point for the static and dynamic analysis is the driving time estimation for ambulances driving with blue lights and siren. The assumption of an overall single average speed is too inaccurate for our purpose. We estimated a specific speed profile for ambulances driving with blue lights and siren based on a large set of real emergency operations and a detailed road network with 20 different road classes.

Speed Profile Estimation In a preprocessing, reliable emergency operations were selected. We used the starting point, the destination and the measured driving time of these emergency operations to estimate an average speed for all road classes with the following iterative process.

1. Set a standard car speed profile as current speed profile.
2. For each emergency operation: Calculate the fastest route from the starting point to the target location using the current speed profile. For this route, calculate the driven distance on each road class.
3. Estimate the speed for each road class using robust linear regression[1] and the original driving time
4. Set the resulting speed profile as current speed profile.
5. Stop if the current speed profile estimates the original driving times with sufficient accuracy. Otherwise go to 2.

For the fulfillment of legal regulations, it is important that all potential rescue locations can be reached within a specific time limit starting from an ambulance station. This motivates the consideration of so-called isochrones.

Isochrones The goal is to calculate the part of the road network which is reachable from an ambulance station within a given time limit. For the transformation of the road network[2] into a graph $G = (V, E)$, we used a tool developed by the Optimization Group at TU Kaiserslautern. In this graph, crossings might be represented by several nodes and edges for modeling driving regulations. We set the edge weight w_e for each edge $e \in E$ as $w_e = l_e \cdot \frac{1}{v_{c(e)}}$, where l_e is the length and $v_{c(e)}$ is the average velocity for the road class $c(e)$ of edge e according to the speed profile.

[1] We used the rlm-function provided by R (http://stat.ethz.ch/R-manual/R-devel/library/MASS/html/rlm.html).

[2] https://www.openstreetmap.de/.

For each ambulance station $x \in X$ we calculate the nearest node $s \in V$ according to the Euclidean distance in the plane. For simplicity, we assume from now on $X \subseteq V$. Starting from node x, we calculate the length $dist(x, v)$ of the shortest path from x to all vertices $v \in V$ using Dijkstra's algorithm. In order to improve the performance, we stop the algorithm as soon as the minimum distance label exceeds a given threshold. We use a threshold of 20 min since in practice each edge must be reachable within 15 min from an ambulance station.

Based on this calculation, we define the *d-isochrone* of $x \in X$ as the induced subgraph $G\left[V_{x,d}\right]$ where $V_{x,d} \subseteq V$ is the subset of nodes $v \in V$ with $dist(x, v) \leq d$. Note that in the literature, a d-isochrone is often modeled as a polygonal subset of the plane. Yet there is no canonical definition of this polygon and it is difficult to give a definition that works reasonable in all cases.

The consideration of an individual isochrone for a station $x \in X$ makes little sense since a vertex might be reachable faster from another station $x' \in X, x \neq x'$. Therefore, we define a *d-isochrone-overlay* as the induced subgraph $G_d = G[V_d]$ where $V_d \subseteq V$ is the subset of nodes $v \in V$ with $\min_{x \in X} dist(x, v) \leq d$. Obviously, it holds that G_d is a subgraph of $G_{d'}$ for all $d \leq d'$. Figure 1 shows the 5-, 10- and 15 min-isochrone-overlay.

Isochrones assume that all ambulances are available at their station. This is not always the case in practice. Ambulances might be occupied, off duty or driving back to their station. Taking these circumstances into account forces a dynamic analysis by discrete event simulation.

Simulation For the dynamic analysis, we estimate the timing of the rescue operations under varying location structures using discrete event simulation. The main events are stated in Table 1 and will be briefly explained in the following.

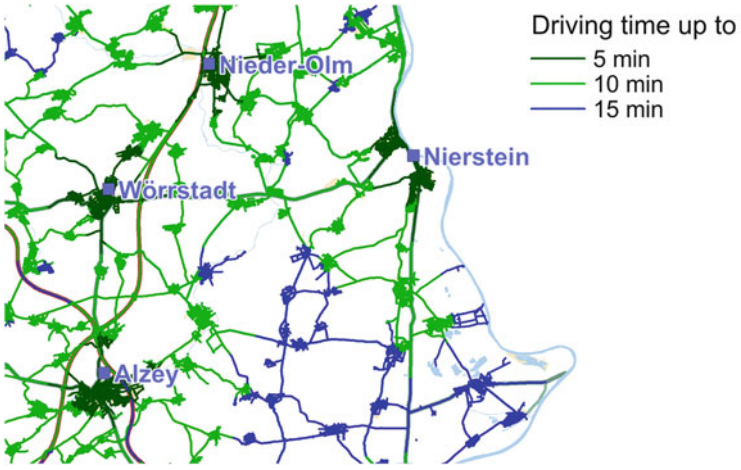


Fig. 1 Isochrone-overlay

Table 1 Events for discrete event simulation

Event	Description	Event time
BeginDuty	Sets vehicle status to *on duty*. Vehicle can be alarmed	given by historic data
CheckEndDuty	Sets vehicle status to *off duty* if the vehicle is at the station. Vehicle can not be alarmed	given by historic data (repeat check later if necessary)
RescOpAlarm	New rescue operation occurs	given by historic data
RescOpDispatch	Assigns the fastest suitable vehicle to the rescue operation. Mark the vehicle as *occupied*	$t_{RescOpAlarm}$ + 1 min
RescOpStart	Assigned vehicle starts to carry out the rescue operation	$t_{RescOpDispatch}$ + 1 min
RescOpEnd	Calculates the route back to the station. Mark the vehicle as *driving empty*	$t_{RescOpStart}$ + estimated driving time + treatment duration (from data)
CheckArrival-AtStation	Marks the vehicle as *waiting* if it arrives at the station. Otherwise it was alarmed already	$t_{RescOpEnd}$ + estimated driving time

We consider for each vehicle individual daily operating hours during those the vehicle can be alarmed. Since a vehicle might still carry out a rescue operation when its operating hours end, we need to check whether it is at the station before it goes off duty. Nevertheless, a vehicle can only be alarmed between the start and end of duty.

A simulated rescue operation is characterized by the alarm time, the rescue location, the original type of the vehicle, the treatment duration and the urgency (whether driving with blue lights and siren or not). In a preprocessing, all historic rescue operations with a sufficient data quality were selected.

The most interesting event is *RescOpDispatch*. It assigns the earliest possible vehicle among all suitable vehicles to the rescue operation. A vehicle a is suitable if it is available (on duty and not occupied) and has a sufficient qualification $q(a)$, i.e. at least as capable as the original vehicle. We distinguish three levels of qualification q_1, q_2, q_3 where $q_1 > q_2 > q_3$. When assigning non-urgent rescue operations, we prefer vehicles with low but sufficient qualification. For all suitable vehicles, we calculate their position on the road network at the time of alarm. Then the driving time of each suitable vehicle from its current position to the rescue location is calculated. The speed profile is dependent on the urgency of the rescue operation. If the historic rescue operation was carried out with blue lights and siren, we use the speed profile according to Sect. 2. Otherwise we use a common car speed profile. The dispatching algorithm is summarized as Algorithm 1.

Algorithm 1 Dispatching algorithm

procedure DISPATCH(RescueOperation $rescOp$)
  Set q as the original vehicle type assigned to $rescOp$
  Set speed profile SP according to the urgency of $rescOp$
  Set $A = \{a \in \text{Vehicles} \mid a \text{ is available and } q(a) \geq q\}$
  if $A = \emptyset$ **then return** $rescOp$ can not be carried out
  end if
  for all vehicles $a \in A$ **do**
    Calculate a's current position, either at the station or by interpolation on the road network
    Calculate a's driving time t_a to the rescue location using SP
  end for
  if $rescOp$ is urgent **then**
    $a^* = \arg\min_{a \in A}(t_a)$
  else
    $a^* = \arg\min_{a \in A}(t_a \mid \text{qualification is minimal})$
  end if
  return new event $RescOpStart(a^*, t_{RescOpStart})$
end procedure

3 Simulation Study and Results

In a real world study, we investigated the location structure of ambulances in the eastern part of Rheinhessen, Germany. As an replacement for the outdated station in Nierstein, we identified together with a team of EMS experts and local politicians seven potential location scenarios with different vehicle configurations. With the static and dynamic analysis we found the best scenario.

For the static analysis, we considered the location of the potentially new station(s) and 15 existing ambulance stations. For each station the 20min-isochrone was calculated using our estimated speed profile for ambulances. For example the 15min-isochrone around the station in Wörrstadt contains 28,790 road segments. After filtering out farm paths, 18.369 road segments remain. This isochrone covers an area of around 520 km^2 with 187.670 habitants. From the 15min-isochrone-overlay, we found that the ambulance from Wörrstadt is the *fastest* vehicle in an area of 150 km^2 with 35.520 habitants. With the static analysis and our estimated speed profile, we identified in each scenario all regions and road segments which are probably not reachable by an ambulance within the legal time limit of 15 min.

We simulated the emergency medical services over a time horizon of almost 4 years. We considered the potentially new station(s) and all 10 ambulance stations in the direct neighborhood with 55 ambulances in total. In a preprocessing, all reliable rescue operations were filtered out of 600.000 data items. For each simulated rescue operation, the simulation calculates up to 55 routes. In order to improve the performance and avoid redundant route calculations, we cache the calculated routes. Especially routes from e.g. an hospital back to a station are used frequently. Depending on memory consumption, the cache might be cleared several times.

The study identified a clearly superior scenario. When establishing the new station again in Nierstein and an additional station with a single ambulance 8km

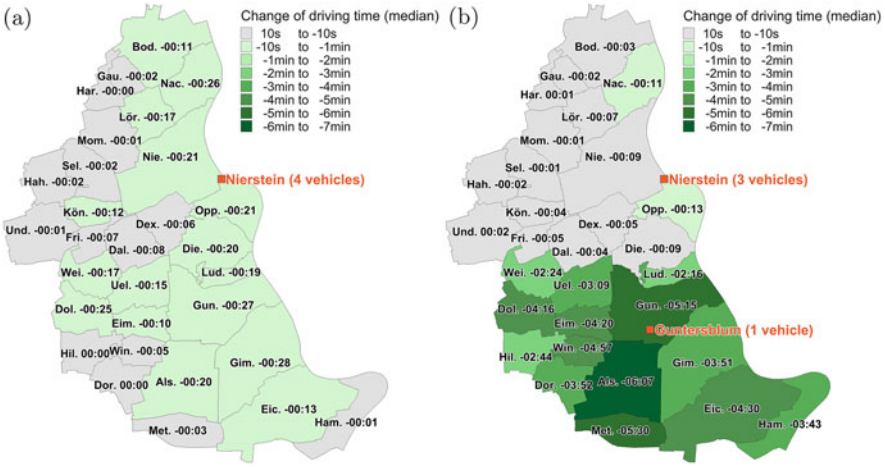


Fig. 2 Change of the driving time median for the investigated municipalities. (**a**) An additional vehicle in Nierstein slightly improves the driving times. (**b**) Moving a vehicle to Guntersblum improves the driving times enormously

south of Nierstein, the median of the driving time to municipalities in the southern part of the investigated area improves between 2 and 6 min. The northern part of the investigated area is mainly unchanged. In contrast to that, an additional vehicle in Nierstein would only lead to an improvement of the driving time median of at most 28 s. See Fig. 2 for a comparison of the driving time median for these two scenarios.

4 Conclusion and Future Work

In this paper, we presented a static and dynamic analysis of ambulance stations based on an estimated speed profile. Isochrones were used to detect areas which are reachable within 15 min from a station. The dynamic analysis uses discrete event simulation for realistic estimation of driving times and workloads. These techniques were used for a real world study and proved a superior scenario.

In the project HealthFaCT, we are extending the methods for larger areas and apply them to other branches of the health care system such as pharmacies.

References

1. Aboueljinane, L., Sahin, E., Jemai, Z.: A review on simulation models applied to emergency medical service operations. Comput. Ind. Eng. **66**(4), 734–750 (2013). https://doi.org/10.1016/j.cie.2013.09.017

2. Lutter, P., Degel, D., Wiesche, L., Werners, B.: Analysis of Ambulance Location Models Using Discrete Event Simulation. Operations Research Proceedings 2014, pp. 377–383. Springer International Publishing (2016). https://doi.org/10.1007/978-3-319-28697-6_53
3. Aboueljinane, L., Jemai, Z., Sahin, E.: Reducing ambulance response time using simulation: the case of Val-de-Marne department Emergency Medical service. Proceedings - Winter Simulation Conference (2012). https://doi.org/10.1109/WSC.2012.6465018
4. Stämpfli, A., Strauss, C.: sim911 - Ein Simulator für das Rettungswesen https://www.fhsg.ch/fhs.nsf/files/IMS_Rettungswesen_sim911Bericht/$FILE/1-sim911-EinSimulatorfu%CC%88rdasRettungswesen.pdf

Assigning Students to Schools for an Internship

Michael Stiglmayr, Simon Görtz, and Kathrin Klamroth

1 Introduction

Teachers education in Germany is a two-stage process. After graduating with the degree "Master of Education", prospective teachers complete an 18 month traineeship at a school (Referendariat). In order to additionally strengthen the practical aspects already during the Master of Education program, an internship at a school is mandatory at all universities in North-Rhine-Westphalia since the summer term 2015 [5]. To manage the administrative effort associated with this newly introduced internship, the universities in North-Rhine-Westphalia involved in teachers education developed a web-based framework called PVP.[1] It was developed to manage and moderate the interests of students, schools and universities, and at its heart is an optimization module that was developed at the University of Wuppertal. Using web-interfaces, the associated interest groups (i.e., universities, students, schools and the ministry of education) can edit their data-sets, specifying, e.g., capacity constraints at schools (schools) or indicate preferred schools (students). When all data is collected, then university administrators start the optimization module and use the obtained results to finalize and communicate the assignment of students to schools.

[1] PVP website (German) https://www.pvp-nrw.uni-wuppertal.de/ueber-pvp.html.

M. Stiglmayr (✉) · K. Klamroth
University of Wuppertal, School of Mathematics and Natural Sciences, Wuppertal, Germany
e-mail: stiglmayr@math.uni-wuppertal.de; klamroth@math.uni-wuppertal.de

S. Görtz
University of Wuppertal, Administration Department of Studies, Teaching and Quality Management, Wuppertal, Germany
e-mail: goertz@uni-wuppertal.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_20

2 Integer Programming Model

Let $\mathcal{I} = \{1, \ldots, I\}$ represent the set of students, and let $\mathcal{J} = \{1, \ldots, J\}$ represent the set of schools. Furthermore, denote by $\mathcal{K} = \{1, \ldots, K\}$ the set of subjects and let $\alpha_{i,k}$ be a parameter indicating the subject combination of student $i \in \mathcal{I}$, i.e. $\alpha_{i,k} = 1$ if student i studies subject k and zero otherwise. Usually, every student studies two subjects, so there are only two '1' entries in every row of α. On the other hand, the schools have subject depending capacity bounds, since the supervision of student trainees is in the responsibility of the teachers at the school. Therefore, let $\beta_{j,k}$ be the capacity of school j in subject k. A student can only be assigned to a school if it provides capacity in both of its subjects.

Every student can choose up to five preferred schools and rank them according to his or her preferences. In addition, every student identifies a reference location within North-Rhine Westphalia (usually this will be the place where he or she lives). We then compute a rating $c_{i,j} > 0$ for the assignment of student i to school j based on this ranking. Non-ranked schools receive a rating based on the reciprocal distance from the reference location. Thereby we want to ensure that the travel time is as small as possible if the student is not assigned to a preferred school. Then the problem of assigning students to schools can be modeled as a binary linear programming problem with decision variables $x_{i,j} \in \{0, 1\}$, where $x_{i,j} = 1$ indicates that student i is assigned to school j, as follows:

$$\begin{aligned} \max \quad & \sum_{i=1}^{I} \sum_{j=1}^{J} c_{i,j}\, x_{i,j} \\ \text{s.t.} \quad & \sum_{j=1}^{J} x_{i,j} = 1 && \forall i \in \mathcal{I} \\ & \sum_{i=1}^{I} \alpha_{i,k}\, x_{i,j} \le \beta_{j,k} && \forall j \in \mathcal{J},\ k \in \mathcal{K} \\ & x_{i,j} \in \{0, 1\} && \forall i \in \mathcal{I},\ j \in \mathcal{J}. \end{aligned} \tag{1}$$

The constraints guarantee that every student is assigned to exactly one school, and that the capacities at the schools are not exceeded in any subject.

Proposition 1 *Model* (1) *is in general not totally unimodular.*

Proof Consider the system of linear inequalities of the problem (1).

$$\begin{pmatrix} \boxed{\begin{matrix} \alpha_{1,1} & \ldots & \alpha_{I,1} \\ \vdots & & \vdots \\ \alpha_{1,K} & \ldots & \alpha_{I,K} \end{matrix}} & & \\ & \boxed{\begin{matrix} \alpha_{1,1} & \ldots & \alpha_{I,1} \\ \vdots & & \vdots \\ \alpha_{1,K} & \ldots & \alpha_{I,K} \end{matrix}} & \\ & & \ddots \end{pmatrix} \cdot \begin{pmatrix} x_{1,1} \\ \vdots \\ x_{I,1} \\ x_{1,2} \\ \vdots \\ x_{I,2} \\ \vdots \\ x_{I,J} \end{pmatrix} \le \begin{pmatrix} \beta_{1,1} \\ \vdots \\ \beta_{1,K} \\ \beta_{2,1} \\ \vdots \\ \beta_{2,K} \\ \vdots \\ \beta_{J,K} \end{pmatrix}$$

Then the matrix

$$\tilde{\alpha} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

may be a submatrix of $\alpha = (\alpha_{i,k})_{i\in\mathcal{I},k\in\mathcal{K}}$. But the determinant of this submatrix is $\det(\tilde{\alpha}) = 2$. Thus, model (1) is in this case not TU.

Example 1 Based on the proof of Proposition 1, one can easily construct instances such that the corresponding LP-relaxation has a fractional optimal solution. Consider the example illustrated in Fig. 1, with $I = 3$, $J = 2$, $K = 3$, $\alpha = \left(\begin{smallmatrix} 1 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{smallmatrix}\right)$, $c = (c_{i,j})_{i\in\mathcal{I},j\in\mathcal{J}} = \left(\begin{smallmatrix} 5 & 1 \\ 4 & 1 \\ 3 & 1 \end{smallmatrix}\right)$ and $\beta = (\beta_{j,k})_{j\in\mathcal{J},k\in\mathcal{K}} = \left(\begin{smallmatrix} 1 & 1 & 1 \\ 2 & 2 & 2 \end{smallmatrix}\right)$. The optimal solution for the integer problem (1) is

$$x^* = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 1 \end{smallmatrix}\right), \qquad \sum_{i=1}^{3}\sum_{j=1}^{2} c_{i,j}\, x^*_{i,j} = 5 + 1 + 1 = 7,$$

whereas the optimal solution of the LP-relaxation is

$$x^{\mathrm{LP}} = \tfrac{1}{2}\left(\begin{smallmatrix} 1 & 1 \\ 1 & 1 \\ 1 & 1 \end{smallmatrix}\right), \qquad \sum_{i=1}^{3}\sum_{j=1}^{2} c_{i,j}\, x^{\mathrm{LP}}_{i,j} = \frac{1}{2}\big(5 + 4 + 3 + 1 + 1 + 1\big) = 7.5$$

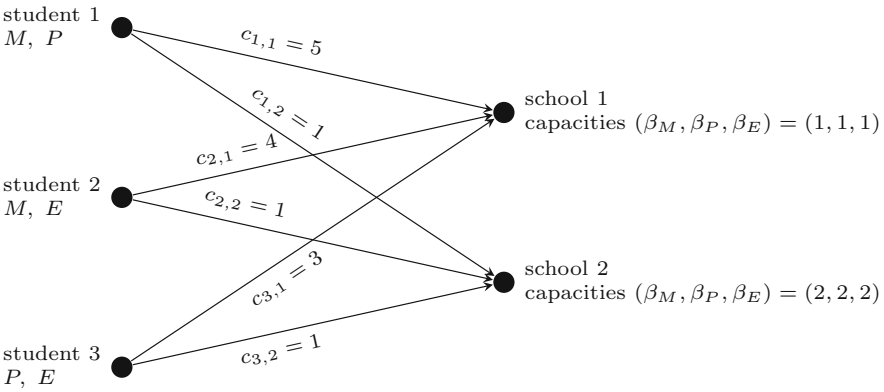


Fig. 1 Example for a non-integral optimal solution. Consider an example with three students with major subjects Mathematics (M), Physics (P) and English (E), and two schools with capacities $\beta_M,\ \beta_P,\ \beta_E$ in these subjects

Formulation (1) is incomplete since, in parallel to their internship, students have to participate in seminar courses at so-called "centers for practical teachers education" ZfSL.[2] Usually, several schools are associated with one ZfSL, and this assignment is fixed. Students completing their internship at a specific school j are automatically allocated to the corresponding ZfSL. Let $\mathcal{M} = \{1, \ldots, M\}$ represent the set of ZfSLs, and denote by $\zeta_{j,m} \in \{0, 1\}$ a binary parameter, with $\zeta_{j,m} = 1$ indicating that school j is associated with ZfSL m. A ZfSL offers seminar courses in different subjects, and each seminar course has a certain subject dependent capacity $\Gamma_{k,m} \geq 0$. The opening costs for a seminar course at a ZfSL is given by $\tau_{k,m}$, $k \in \mathcal{K}, m \in \mathcal{M}$. Since an assignment with many small seminar courses induces much higher costs than an assignment that utilizes fewer but larger seminar courses, we suggest to incorporate the cost for seminar courses into the model. For this purpose, we introduce an additional binary variable $z_{k,m} \in \{0, 1\}$, with $z_{k,m} = 1$ representing the decision to open a seminar in subject k at ZfSL m. This implies that the schools associated with ZfSL m can in total provide at most $\Gamma_{k,m}$ positions for internships in subject k, assuming that the seminar course in subject k is open at ZfSL m.

Furthermore, capacities B_j are introduced limiting the total number of students per school (independent of the subject) to avoid that some schools have a too high workload in organizing the internship.

$$
\begin{aligned}
\max \sum_{i=1}^{I}\sum_{j=1}^{J} c_{i,j}\, x_{i,j} - \sum_{k=1}^{K}\sum_{m=1}^{M} \tau_{k,m}\, z_{k,m} & \\
\text{s.t.} \quad \sum_{j=1}^{J} x_{i,j} &= 1 && \forall i \in \mathcal{I} \\
\sum_{i=1}^{I} x_{i,j} &\leq B_j && j \in \mathcal{J} \\
\sum_{i=1}^{I} \alpha_{i,k}\, x_{i,j} &\leq \beta_{j,k} && \forall j \in \mathcal{J},\ k \in \mathcal{K} \\
\sum_{i=1}^{I}\sum_{j=1}^{J} x_{i,j}\, \zeta_{j,m}\, \alpha_{i,k} &\leq \Gamma_{k,m}\, z_{k,m} && \forall k \in \mathcal{K},\ m \in \mathcal{M} \\
x_{ij} &\in \{0, 1\} && \forall i \in \mathcal{I},\ j \in \mathcal{J} \\
z_{k,m} &\in \{0, 1\} && \forall k \in \mathcal{K},\ m \in \mathcal{M}
\end{aligned}
\tag{2}
$$

[2] Zentrum für schulpraktische Lehrerausbildung.

3 Solution Approaches

3.1 Multicommodity Flow Formulation

Considering the subjects of students as commodities, one can formulate the optimization problem (2) in terms of a multicommodity flow problem [1]. Students, schools and ZfSLs are nodes in a layered digraph, with arcs from all student nodes to all school nodes (modelling the assignment part of the problem) and arcs from schools to their corresponding ZfSLs (reflecting the fixed allocation of schools to ZfSLs). Student node i has a supply vector $\alpha_{i,\bullet}$ wrt. the different commodities $k = 1, \ldots, K$. In order to assign each student with both of its subjects to the same school, we have to ensure that exactly n edges between student and school nodes have a non-zero flow. This can be modeled using additionally to the flow cost $c_{i,j}$ a sufficiently high fixed-charge cost for using this edge. The fixed-charge problem was first proposed in [3] in the context of general linear programming. Fixed-charge network flow also in presence of multiple commodities are presented in [2, 4, 6–8]. Similarly, the edges from the ZfSLs to the sink node are associated with fixed costs $\tau_{k,m}$ (however, there are no flow-dependent costs on these edges). Also in contrast to the classical multicommodity network flow problem we have to use commodity dependent capacity constraint $\beta_{j,k}$ and total capacity constraints B_j on the edges between school nodes and ZfSL nodes. See Fig. 2 for an illustration.

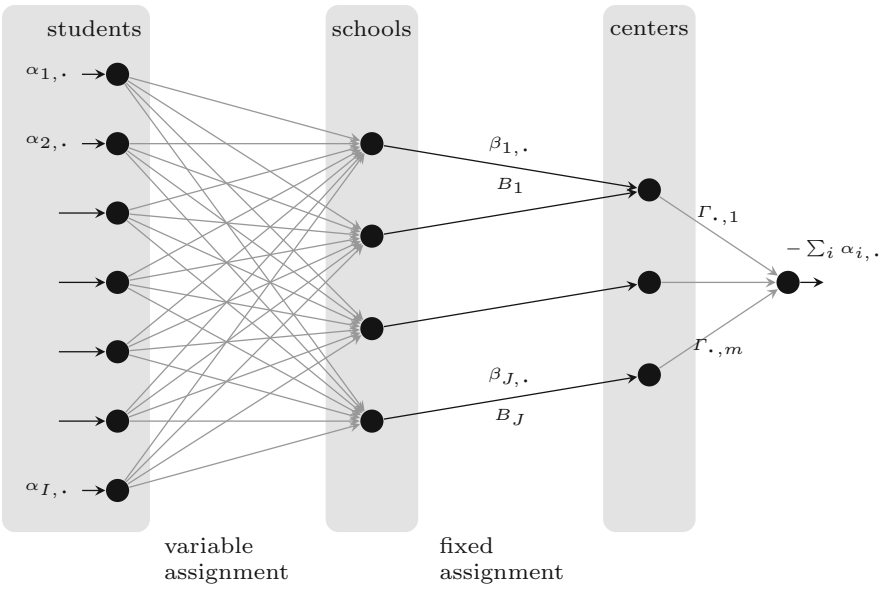


Fig. 2 Fixed-charge multicommodity flow formulation

3.2 Branch and Cut Solution with Cbc

We use the COIN-OR branch and cut solver Cbc[3] to solve model (2) to optimality. Extensive computational experiments on over 400 real-world data sets with up to 298 students studying 34 subjects, 415 schools, and 4 ZfSLs show that even large-size problem instances can be solved to global optimality within 10 min on an Intel Xeon CPU E5520@2.27GHz. However, the majority of instances are in the size of 50–150 students and can be solved to optimality in less than 20 s. The average computation time of all real world instances of the last 5 years is 27.5 s.

Note, that the ratio of values in c and τ has a strong impact on the computation times. Model (2) is considerably harder to solve if both parts of the objective function $\sum_{i=1}^{I}\sum_{j=1}^{J} c_{i,j}\, x_{i,j}$ and $\sum_{k=1}^{K}\sum_{m=1}^{M} \tau_{k,m}\, z_{k,m}$ are balanced and it is easier to solve if one part dominates the other.

4 Conclusion

The problem of optimizing the assignment of students, studying to become teachers, to schools for their internships is modelled as an integer linear programming problem. The resulting generalized assignment problems can be interpreted as fixed-charge multicommodity flow problems, and they can be solved to global optimality using standard IP solvers. The system was implemented in the state of North-Rhine-Westphalia in Germany in a web-based tool and runs successfully since 2012.

References

1. Assad, A.: Multicommodity network flows – a survey. Networks **8**, 37–91 (1978)
2. Chouman, M., Crainic, T.G., Gendron, B.: A branch-and-cut algorithm for multicommodity capacitated fixed charge network design. Technical report, Université de Montréal (2008)
3. Hirsch, W.M., Dantzig, G.B.: The fixed charge problem. Naval Res. Logist. Q. **15**(3), 413–424 (1968). https://onlinelibrary.wiley.com/doi/abs/10.1002/nav.3800150306
4. Kim, D., Pardalos, P.M.: A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. Oper. Res. Lett. **24**(4), 195–203 (1999)
5. Ministerium für Schule und Bildung des Landes Nordrhein-Westfalen: Das Bildungsportal: Lehrkraft in Nordrhein-Westfalen (2018). https://www.schulministerium.nrw.de/docs/LehrkraftNRW/Lehramtsstudium/Praxiselemente/Praxissemester/index.html. Accessed 09 Jul 2018
6. Murty, K.G.: Solving the fixed charge problem by ranking the extreme points. Oper. Res. **16**(2), 268–279 (1968). http://www.jstor.org/stable/168755

[3] https://projects.coin-or.org/Cbc.

7. Nahapetyan, A.G., Pardalos, P.M.: Adaptive dynamic cost updating procedure for solving fixed charge network flow problems. Comput. Optim. Appl. **39**(1), 37–50 (2008). http://dx.doi.org/10.1007/s10589-007-9060-x
8. Ortega, F., Wolsey, L.A.: A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. Networks **41**(3), 143–158 (2003). http://dx.doi.org/10.1002/net.10068

Part V
Energy and Environment

Electricity Price-Oriented Scheduling Within Production Planning Stage

Jan Busse and Julia Rieck

1 Introduction and Problem Specification

The industrial sector utilizes a variety of energy sources, e.g., electricity, natural gas, petroleum, and coal. Particularly, electricity is used for operating industrial machines, motors, computers, and lights. In case of purchase of electricity, production companies are faced with volatile prices (as a result of integrating more and more renewable sources like wind and solar in the electricity production). The price volatility causes the implementation of electricity savings strategies and a comprehensive energy management. Consequently, manufacturing companies observe and assess the daily trade of electricity in order to schedule energy-intensive jobs in time intervals of low prices.

In what follows, we focus on a flow shop scheduling problem, where jobs have to pass a system of up to three stages (machines). Every job $j \in \mathcal{J}$ has a given processing time $tp_{js} \in \mathbb{Z}_{\geq 0}$ on stage $s \in \mathcal{S}$ that has to be carried out without interruption. Between stages the execution of a job can temporarily be paused (cf. the grey space in Fig. 1a). During execution, a job requires energy and the energy demand varies during processing. However, preliminary studies showed that the energy demand is rather constant in specific production phases (micro steps $m \in \mathcal{M}$), cf. [5]. Typically, a starting phase, a shutdown phase and one or two execution phases may be determined. Figure 1b shows an energy load profile of job j on stage one. The fluctuating energy demand is visualized with a solid line and the constant energy demand in micro steps $m = 1, 2, 3$ is given with a dashed line. $tp_{jsm} \in \mathbb{Z}_{\geq 0}$ refers to the processing time of job j on stage s in micro step

J. Busse (✉) · J. Rieck
Operations Research Group, Institute of Economics and Computer Science,
University of Hildesheim, Hildesheim, Germany
e-mail: busse@bwl.uni-hildesheim.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_21

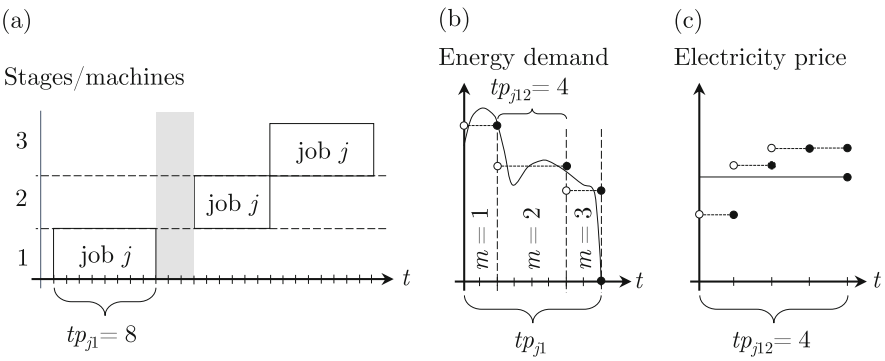


Fig. 1 (**a**) Possible schedule for job j, (**b**) load profile on stage $s = 1$ and (**c**) average electricity price on stage $s = 1$, micro step $m = 2$

m. In our study, a mid-term planning horizon $\mathcal{T}$ of one week is considered, where hourly electricity prices can vary by up to 30€/MWh. We assume that the planning horizon is divided into equidistant time points $t \in \mathcal{T}$, where $\frac{tp_{jsm}}{\alpha} \in \mathbb{Z}_{\geq 0}$ is always satisfied; α is the period length, i.e., the difference between two consecutive time points. Depending on the industrial application, the period length α may be shorter or longer than an hour. Since the electricity price is on an hourly basis, a method for considering the right price for a time point must be identified. We assume that an hour contains $\frac{1}{\alpha} = \beta$ time points. In the event that $\beta \geq 1$ (period length is less than an hour), the underlying hourly price P_h determines the electricity price for each inner time point (namely, $t^1, \ldots, t^\beta$), i.e., we set $p_{t^1} = \ldots = p_{t^\beta} := P_h$. In case $\beta < 1$ (period length is greater than an hour), we assign the average electricity price to the corresponding time point. Assuming that $P_{h^1}, \ldots, P_{h^\alpha}$ are hourly prices that occur between time points $t - 1$ and t, then, p_t may be calculated as follows $p_t = \frac{1}{\alpha} \sum_{i=1}^{\alpha} P_{h^i}$, cf. the solid line in Fig. 1c. The objective of our problem is to minimize the total electricity costs arising from the job-specific energy demand and the hourly varying electricity prices within $\mathcal{T}$.

In a tactical planning environment, where jobs have to be scheduled at low prices, production companies need information about electricity prices for taking decisions on production. Consequently, there is a need for efficient forecasting techniques that help to estimate the tariff prices before they will be emerged on the market. In order to include reliable energy price information, a day-ahead price forecast was executed taking into account real-time EPEX SPOT prices from recent past. For this purpose, we implemented a feedforward neural network (DFNN) with a forecast period of 4 weeks. In doing so, we computed 1357 different forecasts (day-by-day-shifting of input data) to validate the network's accuracy.

The problem under consideration deals with the shift of jobs (and energy demand, respectively) such that energy costs are minimized and it can be found in common manufacturing environments such as single and parallel machine, job shop and flow shop systems (e.g., [3, 4, 6, 8]). However, mid-term scheduling as well as

forecast considerations are lacking. Besides, many studies on a tactical level focus on the energy consumption which is directly linked to energy costs but without any consideration of volatile electricity prices [1, 2, 7].

2 Model Formulation

In this section, we present a Mixed-Integer Linear Programming (MILP) model for our electricity price-oriented flow shop scheduling problem with s stages in series. All jobs have to follow the same route, allowing to skip a stage if no operation is required. Let $\mathcal{T}$ be the planning horizon (i.e., one week) divided in equidistant time points and let $\mathcal{J}$ be the finite set of jobs to be processed in an energy-intensive production environment (e.g. steel, chemical or aerospace industry).

The problem objective consists of the minimization of the energy costs. Using the variables and parameters given in Table 1, the model has the following form:

$$\min \quad \sum_{j\in\mathcal{J}}\sum_{s\in\mathcal{S}}\sum_{\tilde{m}\in\mathcal{M}}\sum_{t\in\mathcal{T}}\sum_{\tau=t+\sum_{m=1}^{\tilde{m}-1} tp_{jsm}}^{t-1+\sum_{m=1}^{\tilde{m}} tp_{jsm}} p_\tau \cdot ec_{js\tilde{m}} \cdot X_{js,t-1+tp_{js}} \tag{1}$$

$$\text{s.t.} \quad \sum_{t\in\mathcal{T}} X_{jst} = \Theta_{sj} \qquad \forall\, j, s \tag{2}$$

$$TPE_{js} \geq TPE_{j,s-1} + \sum_{m\in\mathcal{M}} tp_{jsm} \cdot \Theta_{\mathrm{sj}} \qquad \forall\, j, s \geq 2 \tag{3}$$

$$TPE_{jsm} = TPE_{jsm-1} + tp_{jsm} \cdot \Theta_{\mathrm{sj}} \qquad \forall\, j, s, m \geq 2 \tag{4}$$

$$TPE_{js} = \sum_{t\in\mathcal{T}} t \cdot X_{jst} + TPE_{j,s-1} \cdot (1 - \Theta_{\mathrm{sj}}) \qquad \forall\, j, s \tag{5}$$

Table 1 Parameters and variables (alphabetically ordered)

Parameters	
ec_{jsm}	Energy demand of job j on stage s in micro step m [MW]
p_t	Electricity price at time t [€/MW]
Θ_{sj}	Stage-indicator, 1 if job j has to be processed on stage s, 0 otherwise
tp_{jsm}	Processing time of job j on stage s in micro step m
Decision variables	
TPE_{js}	Completion time of job j on stage s
TPE_{jsm}	Completion time of job j on stage s in micro step m
X_{jst}	1 if job j is completed on stage s at time t, 0 otherwise

$$TPE_{js} = TPE_{js|\mathcal{M}|} \qquad \forall j, s \tag{6}$$

$$\sum_{j \in \mathcal{J}} \sum_{\tau=t}^{t-1+\sum_{m \in \mathcal{M}} tp_{jsm}} X_{js\tau} \leq 1 \qquad \forall s, t \tag{7}$$

$$X_{jst} \in \{0, 1\} \qquad \forall j, s, t \tag{8}$$

$$TPE_{js} \geq 0 \qquad \forall j, s \tag{9}$$

$$TPE_{jsm} \geq tp_{jsm} \qquad \forall j, s, m \tag{10}$$

The objective function (1) minimizes the total energy costs resulting from the completion times to be determined. Constraints (2) ensure that each job j processed on stage s is executed exactly once. Inequalities (3) model the completion time of job j on stage s which is equal or greater than its completion time on predecessor stage $s-1$ plus a possible total processing time on stage s. Due to inequality conditions, positive transition/transportation times between consecutive stages are possible. Equations (4) link micro step m with its predecessor $m-1$ within stage s, so that m starts immediately after the completion of $m-1$ (without any process interruption). Equalities (5) ensure that the completion time of job j on stage s is defined correctly in the case that j skips stage $s-1$. The completion time of job j on stage s equals the one of the job's last micro step at this stage, cf. conditions (6). At each time point t, capacity utilization of stage s must not exceed the total capacity available, which is one machine on each stage, cf. inequalities (7). Finally, domains of decision variables are declared by constraints (8), (9) and (10).

3 Computational Study

The purpose of the computational study is twofold: (a) to analyze the suitability of long-term hourly electricity price-oriented forecasts for scheduling problems and (b) to explore the potential of energy-cost oriented scheduling compared to common makespan minimization. For our performance analysis, the scheduling model (cf. Sect. 2) was implemented in GAMS and instances were solved to optimality by CPLEX 12.7.1.0 on an Intel i7 CPU with 3.4 GHz and 16 GB RAM.

We used the forecast prices for the first week of June 2016 obtained from our DFNN as input variables. Furthermore, we accounted for the real electricity prices occurred during this period. The computational tests have been performed on 150 randomly generated instances which have been clustered in 15 test sets, each comprised of 10 instances. The data generation process has been derived from real-world data provided by an industry partner. We assume a planning horizon of $|\mathcal{T}| = 168$ h (7 days) and consider different scenarios composed of $|\mathcal{J}| = 10$, $|\mathcal{J}| = 20$ and $|\mathcal{J}| = 30$ jobs. In order to cover a wide range of possible job types with different processing times and variable energy demand on stage s, we combine

Table 2 Scenario construction for stage s

Energy demand [MW]		Processing time [h]	
Case 1	0.1–3	Case a	1–10
Case 2	3–6	Case b	11–20
Case 3	0.1–6	Case c	1–20

Table 3 Computational results for the forecast error and the makespan-comparison for a summer week (2016-06-01 to 2016-06-07), 150 instances

	(a) Forecast				(b) Makespan		
Test set	t_{cpu} [s]	$\varepsilon_{\text{mape}}$ [%]	ε_{mae} [€]	$\varepsilon_{\text{rmse}}$ [€]	Δ [%]	Δ_{abs} [€]	Δ_{quad} [€]
7d-10-1a	13.40	2.95	50.28	52.20	56.05	954.46	977.86
7d-10-1b	22.65	1.55	84.62	91.89	12.30	664.54	678.51
7d-10-1c	17.13	2.22	74.02	77.10	34.60	1152.26	1177.34
7d-10-2a	19.90	3.54	193.72	200.60	47.28	2552.81	2572.47
7d-10-2b	29.48	1.70	274.08	279.90	11.15	1803.60	1838.41
7d-10-2c	23.26	2.22	204.94	214.94	30.32	2800.51	2820.78
7d-10-3a	16.54	3.53	137.49	144.49	53.08	2059.33	2109.06
7d-10-3b	26.20	1.48	157.68	184.97	16.98	1749.39	1779.97
7d-10-3c	21.02	2.32	147.91	160.64	31.03	1928.56	1956.90
7d-20-1a	43.94	2.90	112.52	114.16	37.63	1472.72	1488.08
7d-20-2a	74.13	2.70	325.34	334.28	32.12	3826.49	3850.00
7d-20-3a	44.33	3.18	207.71	210.57	40.94	2691.07	2716.49
7d-30-1a	293.14	2.51	156.93	160.24	23.02	1438.28	1457.15
7d-30-2a	780.83	2.22	412.18	414.03	17.53	3246.44	3262.72
7d-30-3a	446.69	2.41	288.18	292.86	23.19	2717.07	2734.20

various cases as shown in Table 2. The scenario $C = 2\text{a}$, e.g., covers a random energy demand between 3 and 6 MW as well as random processing times between 1 and 10 h at each stage s.

Table 3 shows the computational results, where the different test sets are given by the names 7d-$|\mathcal{J}|$-C to clarify the planning horizon, the number of jobs and the scenario applied. Please note that for $|\mathcal{J}| = 20, 30$ only test sets with short processing times (case a) have been generated, as cases b and c have caused too many infeasible instances.

Column t_{cpu} shows the average computation time for the optimally solved instances of each test sets. Furthermore, the error measures MAPE ($\varepsilon_{\text{mape}}$), MAE ($\varepsilon_{\text{mae}}$) and RMSE ($\varepsilon_{\text{rmse}}$) are listed. Looking at the makespan investigation, column Δ displays the decline in energy costs when minimizing makespan instead of using our objective function. Further distance measures are given by columns Δ_{abs} and Δ_{quad} to capture average absolute deviations of the objective values and to identify potential outliers in the test set. Results show a mean average deviation of the forecasted energy costs from the actually occurred ones ranging from $1.48\% \leq \varepsilon_{\text{mape}} \leq 3.54\%$, which is remarkable. Note that scenarios 1b, 2b and 3b perform best. This can be explained by the long scenario-specific processing times which

hence cover greater forecasted time intervals. Deviations of the forecasted electricity prices from the real values level out here and the possibility to switch jobs to low electricity price intervals is limited. By contrast, the absolute errors ε_{mae} and $\varepsilon_{\text{rmse}}$ are greater due to much longer processing times within the forecast horizon.

The results obtained for the makespan-comparison show that energy costs increase up to 56.05%. Particularly, instances with 10 jobs and short processing times (i.e., scenarios 1a, 2a, 3a) perform worse, since this small amount of jobs could be easily switched to low electricity price intervals within $\mathcal{T}$. The greatest increase in energy costs with up to $\Delta_{\text{abs}} = 3826.49€$ can be found in scenarios with high energy consumption and short processing times (2a).

4 Conclusion

In the paper at hand, a flow shop scheduling model is presented that accounts for hourly varying electricity prices. We conducted an extensive computational study covering a wide range of job structures in order to observe the impact of problem structures. By means of a long-term electricity price forecast we could outline the benefit of applying this forecast for scheduling problems. The average increase in energy costs is less than 3.54% holding for all test sets which might be good news for a production planner. This error even decreases up to 1.48% for long processing times. Furthermore, we showed that the common makespan minimization without regard to electricity prices leads to an increase of energy costs of up to 56%, illustrating the considerable potential of the approach.

References

1. Fang, K., Uhan, N.A., Zhao, F., Sutherland, J.W.: Flow shop scheduling with peak power consumption constraints. Ann. Oper. Res. **206**(1), 115–145 (2013)
2. Lu, C., Gao, L., Li, X., Pan, Q., Wang, Q.: Energy-efficient permutation flow shop scheduling problem using a hybrid multi-objective backtracking search algorithm. J. Clean. Prod. **144**, 228–238 (2017)
3. Moon, J.-Y., Park, J.: Smart production scheduling with time-dependent and machine-dependent electricity cost by considering distributed energy resources and energy storage. Int. J. Prod. Res. **52**(13), 3922–3939 (2014)
4. Moon, J.Y., Shin, K., Park, J.: Optimization of production scheduling with time-dependent and machine-dependent electricity cost for industrial energy efficiency. Int. J. Adv. Manuf. Technol. **68**, 523–535 (2013)
5. Teiwes, H., Blume, S., Herrmann, C., Rössinger, M., Thiede, S.: Energy load profile analysis on machine level. Procedia CIRP **69**(May), 271–276 (2018)
6. Willeke, S., Prinzhorn, H., Stonis, M., Nyhuis, P.: Preconditions for applying an energy price-oriented sequencing rule. Prod. Eng. **12**(1), 73–81 (2018)
7. Yan, J., Li, L., Zhao, F., Zhang, F., Zhao, Q.: A multi-level optimization approach for energy-efficient flexible flow shop scheduling. J. Clean. Prod. **137**, 1543–1552 (2016)
8. Zhang, H., Zhao, F., Fang, K., Sutherland, J.W.: Energy-conscious flow shop scheduling under time-of-use electricity tariffs. CIRP Ann. Manuf. Technol. **63**(1), 37–40 (2014)

Two-Stage Unit Commitment Modeling for Virtual Power Plants

Lars-Peter Lauven

1 Introduction

The continuous expansion of renewable power generation in Germany leads to a variety of new planning problems. The power output of the two largest groups of renewable energy plants, photovoltaics and wind power, is intermittent, which means that other components of the energy system need to be adapted to react to these fluctuations. As power can only be stored to a very limited extent, such adaptations usually include adapting dispatchable power suppliers, customers with flexible demands and available storage options. Dispatchable power plants, flexible customers and storage devices are referred to as "Flexibility Options" (FO) in this context. In the course of the expansion of volatile power generation capacity, an increasing number of concepts and technologies are being considered as FO. As the cost of supplying flexibility differs considerably between these technologies, a methodology is required to assess the economic viability of combining intermittent power plants and FO.

2 Trading Power Generation Flexibility in Germany

A single aggregator, such as a Virtual Power Plant (VPP), often controls numerous Flexibility Options. VPPs aggregate several intermittent power sources and FO to make power generation more predictable and thus reliable. As part of a VPP, intermittent power sources can pass the prequalification that is required for participating

L.-P. Lauven (✉)
Chair of Production and Logistics, University of Göttingen, Göttingen, Germany
e-mail: Lars.lauven@wiwi.uni-goettingen.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_22

in market auctions as well as avoid punishing costs for balancing power in case of faulty production forecasts. In addition to enabling the market participation of intermittent power sources, the aim of a VPP is to apply the available flexibilities to different markets to maximize profit of their operation [1]. In the following two sub-sections, both different kinds of flexibility options and the potential markets are introduced.

The flexibility that is required to balance supply and demand of power can be found in several parts of the power system. While some flexibility also exists in the management of the power grid itself, the most significant FOs are flexible power suppliers, flexible consumers and a diversity of storage systems.

Since the liberalization of the German power sector, several marketplaces have been established to facilitate power trading. Among the most significant marketplaces in this context are the day-ahead auction, intraday auction and the continuous intraday trading of the European Power Exchange (EPEX) and the various forms of ancillary service markets.

To take part in the day-ahead or intraday auctions, bids on an hourly basis must be submitted on the day before physical delivery. Soon after, the market clearing price is announced for each of the 24 h of the following day [2]. Due to the similarity between the two, the intraday auction will not be discussed in further detail in this paper.

In contrast to the two auctions described before, trading can take place until 30 min before physical delivery. Additionally, there is no market clearing price, but 15 min contracts are traded whenever there is a match between supply and demand of power for a specific time window. During intraday trading, it is uncertain whether reasonably priced power offers will be (or become) available in time before the moment of physical delivery. Therefore, contrary to auctions, the continuous trading requires making numerous decisions throughout the day, as bids may become available any time until 30 min before delivery. To our knowledge, no model currently takes the additional possibility to trade in the continuous intraday trading into account.

If individual actors on the power market fail to cover their power requirements in either one of the auctions or through the continuous intraday trading, they need to purchase balancing power from the transmission system operators (TSOs), often at unfavorable prices.

3 Modeling Flexibility Options with Optimization Models

When modeling the optimal use of flexibility on the described power markets, two stage models can be used to represent initial dispatch and redispatch for example on the second day [3]. Contrary to existing models, the following optimization model models a potential market interaction on two markets, both the day-ahead auction and the continuous intraday trading [4]. In an exemplary combined model for a flexible biogas plant and pumped-storage hydroelectricity as supply-side FOs and a

PV plant as an intermittent renewable power producer, it is possible to explore the value of a combined marketing of dispatchable and intermittent power sources.

In the first stage, i.e. as preparation for the day-ahead auction, the output of the PV plant is calculated based on a forecast of solar radiation of the next day [5].

$$\varphi_i^{DA} A \eta \omega = P_i^{PV,DA} \tag{1}$$

where:

φ_i^{DA}	day-ahead forecast for the solar radiation in hour i (kW/m^2)
A	area of the PV plant (m^2)
η	efficiency of converting solar radiation into power (%)
ω	performance ratio of real vs. ideal yield of a PV plant (%)
$P_i^{PV,DA}$	forecasted power from the PV plant in hour i (kW)

The biogas plant's and pumped-storage hydroelectricity's operating hours are chosen with the following two optimization models based on a forecast of the day-ahead prices. For the biogas plant, revenues based on the day-ahead (DA) prices are maximized under the constraints that the biogas, of which there is a constant production $P_{BGP,G}$ that is independent of power generation $P_{BGP,P}$, can be stored, and that the power generator operates within feasible limits.

$$\max \sum_{i=1}^{I} \left(p_i^{DA} P^{BGP,P} x_i \right) \tag{2}$$

$$cap_j^{BGP} = cap_0^{BGP} + P^{BGP,G} \cdot j - P^{BGP,P} \sum_{i=0}^{j} x_i \tag{3}$$

$$cap_{min}^{BGP} \leq cap_j^{BGP} \leq cap_{max}^{BGP} \tag{4}$$

$$0.35 \leq x_i \leq 1 \tag{5}$$

cap_j^{BGP}	state of biogas storage capacity in time slot j
p_i^{DA}	day-ahead power price in time slot i
$P^{BGP,G}$	capacity of the biogas production
$P^{BGP,P}$	capacity of power generation

For the pumped-storage hydroelectricity model, a term for the value of stored electricity is added. This term is meant to avoid that the energy stored at the beginning of the planning horizon is depleted even when power prices are low. In

contrast to the biogas model, the value of stored energy only grows if pumps fill the reservoir with water (i.e. $y_i < 0$).

$$max \sum_{i=1}^{I} \left(\left(p_i^{DA} - p_{I+} \right) P^{PSH} y_i \right) \tag{6}$$

$$cap_j^{PSH} = cap_0^{PSH} - P^{PSH} \cdot \sum_{i=0}^{j} y_i \tag{7}$$

$$cap_{min}^{PSH} \leq cap_j^{PSH} \leq cap_{max}^{PSH} \tag{8}$$

$$-1 \leq y_i \leq 1 \tag{9}$$

cap_j^{PSH} state of pumped-storage capacity in time slot j
P^{PSH} capacity of the pumped-storage hydroelectricity

Based on the estimation of PV output and the results of the biogas plant and pumped-storage hydroelectricity optimization based on the day-ahead prices, the total bids for the day-ahead auction can be calculated. If it is assumed that due to an accurate forecast of day-ahead prices, the bids in each hour i ϵ I, $P_{VPP,DA,i}$, are successful in the day-ahead auction, these amount to:

$$P_i^{VPP,DA} = P_i^{PV,DA} + P^{PSH} y_i^{DA} + P^{BGP,P} x_i^{DA} \tag{10}$$

In the second stage, i.e. the morning following the day ahead auction, the difference between forecasted and actual power generation of the combined VPP consisting of biogas plant, pumped-storage hydroelectricity and PV plant, can be traded in the continuous intraday (ID) trading. The output of the biogas and pumped-storage hydroelectricity plants can be used to offset errors in the solar radiation forecast. This second stage optimization model is subject to the combined constraints of the individual optimization models in the first stage, i.e. (3–5) and (7–9), as well as to a new constraint that ensures that the production in the second stage at least fulfils the successful bids on the day-ahead auction:

$$P_i^{VPP,DA} + R_i = P_i^{VPP,ID} = P_i^{PV,ID} + P^{PSH} y_i^{ID} + P^{BGP,P} x_i^{ID} \tag{11}$$

R_i Difference between intraday and day-ahead production values

The objective function of the combined model is thus:

$$max \sum_{i}^{I} \left(p_i^{DA} P_i^{VPP,DA} + p_i^{ID}{}_{ID,i} \left(P_i^{VPP,ID} - P_i^{VPP,DA} \right) - p_{I+} P^{PSH} y_i \right) \tag{12}$$

4 Application, Results and Discussion

This combined optimization model (Eqs. 2, 6 and 12) is applied on an exemplary day on which the solar radiation is assumed to be greater than anticipated before the day-ahead auction. As deviations between forecasts and actual prices are hardly available, we show the functionality of the algorithm with a hypothetical case study day. The VPP's assumed characteristics are shown in Table 1.

On this hypothetical day, we assume that both the PV production of the VPP would be greater than the bids on the day-ahead market and that power prices around noon in the continuous intraday trading turned negative (to −99 €/MWh, see Fig. 1). At such prices, it would be very disadvantageous to sell power on the market.

If it is assumed that this surplus must be sold in the continuous intraday trading, the PV plant alone would face negative revenues of −71.3 € on that day. Adding a biogas or pumped-storage hydroelectricity plant results in positive revenues of 614.64 € and 596.89 €, respectively. A VPP consisting of all three plants results in revenues of 1282.83 € and significantly reduces the power produced in the critical hours around noon/ midday compared to the day-ahead auction. However, if sufficient power volumes can still be traded on the intraday market, these values

Table 1 Parameter values for the exemplary VPP

Parameter	Value	Parameter	Value
A	36176 m^2	cap_{min}^{BGP}	0 (MWh_{el} equiv.)
η	13.8%	cap_{max}^{BGP}	4 (MWh_{el} equiv.)
ω	80%	P^{PSH}	1 MW
$P^{BGP,G}$	1 MW	cap_{min}^{PSH}	0 (MWh_{el} equiv.)
$P^{BGP,P}$	0.5 MW	cap_{max}^{PSH}	8 (MWh_{el} equiv.)
		p_{I+}	35 €/MWh

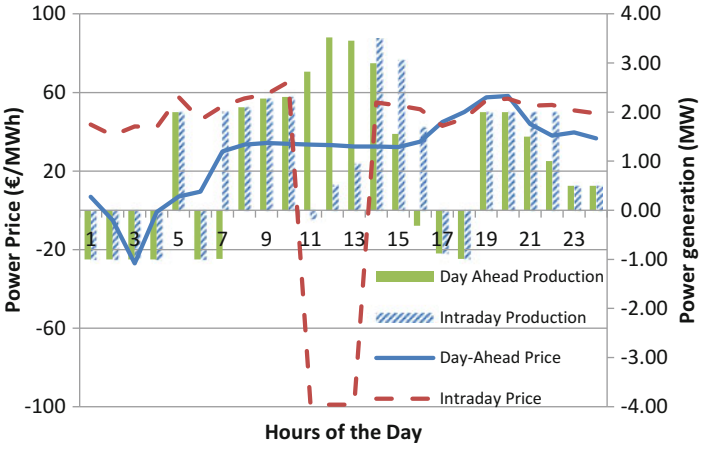


Fig. 1 Unit commitment of the VPP on day-ahead and intraday markets

exactly equal the sum of the revenues of the PV plant and the other two plants if they participated in both day-ahead auction and intraday trading individually. While a redispatch clearly makes sense under these circumstances, the common marketing within a Virtual Power Plant only adds utility to all participants if an internal reaction helps avoid e.g. high prices for balancing power. If the response is measured against a liquid intraday market (in which all kinds of FO can participate), the developed methodology shows that VPPs represent a zero-sum game given that all aggregated FOs can buy and sell unlimited quantities of power in that market. The benefit of aggregating dispatchable and intermittent power plants becomes more decisive when such trading quantities are not available in an hour j, or several hours. In the developed model, this can be included by adding a term to the objective function that subtracts the value of R_j multiplied by the cost of balancing power in hour j from the revenues earned in day-ahead auction and intraday trading. If the cost of balancing power is greater than the shadow price of restricting R_j, the optimal intraday schedule of the VPP changes accordingly.

5 Conclusion and Outlook

The presented methodology can be applied to react to deviations between forecasted and real production values, as well as between day-ahead and intraday prices. It becomes apparent that combining intermittent and dispatchable power plants into an aggregated virtual power plant does not result in immediate gains in trading revenues if sufficiently liquid markets are assumed. The benefit of aggregation for PV plant owners lies in the more reliable power output (i.e. lower cost for balancing power) and in passing the pre-qualification that is required for participation for power trading. For owners of dispatchable power plants, the benefit of participating in VPPs is in the service of marketing their plants on markets, which is increasingly necessary to receive sufficient income for an economic operation.

Further research could include stochastic or robust optimization approaches to deal with deviations between forecasted and real power production from intermittent sources and the effects of such deviations on intraday prices.

References

1. Helms, T., Loock, M., Bohnsack, R.: Timing-based business models for flexibility creation in the electric power sector. Energy Policy. **92**, 348–358 (2016). https://doi.org/10.1016/j.enpol.2016.02.036
2. EPEX Spot SE: Market Data. Day-ahead Auction. http://www.epexspot.com/en/market-data/dayaheadauction/auction-table (2018). Accessed 12 April 2018
3. Huang, Y.: Springer Briefs in Energy. Electrical Power Unit Commitment: Deterministic and Two-Stage Stochastic Programming Models and Algorithms, Springer US (2017)

4. Hospes, Y.: Untersuchung des wirtschaftlichen Potentials der gemeinsamen Vermarktung von regelbaren und nicht regelbaren Erneuerbaren Energien [Investigation of the Economic Potential of the Joint Marketing of Programmable and non-Programmable Renewable Energy], Master Thesis, University of Göttingen (2018)
5. Konstantin, P.: Praxisbuch Energiewirtschaft. [Practical Handbook Energy Economics], 3rd edn. Springer, Berlin Heidelberg (2013)

Globally Optimal Short-Term Unit Commitment and Dispatch for Combined Heat and Power Generation Units Connected to District Heating Grids

Lennart Merkert and Sören Hohmann

1 Introduction

With more and more volatile energy markets due to a raising share of renewable generation, the traditionally heat driven operations strategy of combined heat and power plants (CHPs) is offering a high potential for improvement. However, thermal storage is needed to allow a more flexible operation. Additional to dedicated heat storage tanks requiring an investment of about 1000 €/MWh [1], the thermal inertia of a heating grid can be used as storage avoiding such investment. Nevertheless, optimal operation considering thermal inertia of a heating grid is challenging, as the resulting model formulation is non-convex containing bilinear terms and variable time delays.

In the past there have been several approaches to find suitable solutions for this problem. Many assume constant mass flows and hence time delays to get a linear problem [2, 3]. Others use a linearization (Taylor series) of the non-linear problem [4] or a combination of the two [5]. A rather unique approach was chosen by Groß training a linear regression model with simulations to get a linear representation of the thermal inertia [6]. It is quite common to update the parameters of the linear optimization model iteratively using a non-linear simulation model [3, 5]. The opposite is possible as well. Li et al. propose to repeatedly solve a non-linear optimization with fixed complicating variables (integers) which are updated at each iteration [7]. Another approach uses a sequential solution method based on a simulation model, a linearized unit commitment model and a non-linear

L. Merkert (✉)
ABB Corporate Research Center Germany, Ladenburg, Germany
e-mail: lennart.merkert@de.abb.com

S. Hohmann
Karlsruhe Institute of Technology, Institute for Control Systems, Karlsruhe, Germany

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_23

dispatch model [8]. A drawback of these approaches is that they do not guarantee convergence to a global optimum and hence quality of the solution is unknown.

Global optimization of non-convex problems with bilinear terms is a well-studied field. Many methods are based on McCormick envelopes [9] the tightest convex hull of a bilinear term. We choose to use multiparametric disaggregation [10] for bilinear terms, as it offers faster solving times than a piecewise-linear approximation with McCormick envelopes [11].

Modeling of a delay within a global optimization problem is less well studied. In this paper we are proposing "multiparametric delay modeling" which builds a convex hull for the transport delay and with multiparametric disaggregation allows to find a global optimum for joint unit commitment and dispatch of CHPs connected to heating grids.

2 Modelling of District Heating Grids

A district heating grid consists of generators, pipes and consumers. Generators and consumers are located at nodes which are connected by pipes. In real world installations there is one pipe network for supply and one for return flow. Though as the temperature in the return lines cannot be controlled by the grid operator and hence not be used for heat storage, they are neglected, and we only include supply lines into the model.

Assuming no leakages, the mass flows $\dot{m}_{i,t}$ in and out of one node or pipe are equal.

$$\sum\nolimits_{i \in inflow} \dot{m}_{i,t} = \sum\nolimits_{i \in outflow} \dot{m}_{i,t} \tag{1}$$

With perfect mixing of temperatures, the outflow temperature T_t^{out} of nodes depends on the inflowing mass flows $\dot{m}_{i,t}$ and their temperatures $T_{i,t}^{in}$:

$$T_t^{out} = \frac{\sum_{i \in inflow} T_{i,t}^{in} \cdot \dot{m}_{i,t}}{\sum_{i \in inflow} \dot{m}_{i,t}} \tag{2}$$

The outflow temperature T_t^{out} of a pipe depends on losses influenced by ambient temperature T^{amb} and pipe parameters, the inflow temperature T_t^{in} and a time delay τ (3) [3] which can be calculated based on the velocity v_t of the transport medium (4) [5].

$$T_t^{out} = T^{amb} + \left(T_{t-\tau}^{in} - T^{amb}\right) e^{-\frac{4\,k_v}{c_p \rho \cdot d}\tau} \tag{3}$$

$$\frac{d\tau}{dt} = 1 - \frac{v_t}{v_{t-\tau}} \tag{4}$$

The change of inner energy Q_t of the transport medium in a generator or consumer is

$$Q_t = c_p \dot{m}_t \left(T_t^{supply} - T_t^{return} \right). \tag{5}$$

3 Global Optimization

The district heating grid model introduced in Sect. 2 is non-convex as there are bilinear terms as well as a variable dependent time delay. For the global optimization of bilinear terms multiparametric disaggregation (see 3.1) is used. To globally optimize a variable dependent time delay multiparametric delay modelling is introduced in 3.2.

3.1 Multiparametric Disaggregation

Multiparametric disaggregation has been introduced in [10]. It allows to globally optimize bilinear problems with an iterative optimization scheme by discretizing one variable of the product $w_{i,j} = x_i \cdot x_j$.

$$x_j = \sum_{l=p}^{P} \sum_{k=0}^{9} 10^l \cdot k \cdot z_{jkl} + \Delta \mathrm{x_j}; \sum_{k=0}^{9} z_{jkl} = 1 \tag{6}$$

P and p are defining the precision of the approximation and the slack variable Δx_j leads to a convex hull of the bilinear term. With some additional constraints (see [10]), we get a lower bounding (LB) mixed integer linear program (MILP) of the original problem. Using the non-linear model as upper bound (UB) problem we can reach the global optimum with the following iterative scheme.

1. Solve lower bounding MILP problem (LB problem)
2. Initialize upper bounding non-linear problem (UB problem) with LB solution
3. Solve UB problem (with fixed unit commitment) using a non-linear (local) solver
4. Check gap between UB and LB solution
   a. Gap too big: decrease p (increased accuracy of LB problem) and restart with 1.
   b. Gap ok: globally optimal solution is found

3.2 *Lower Bound of T_t^{out} with Mass Flow Dependent Time Delay*

To find an outer approximation of the pipe outflow temperature T_t^{out} (3) the case shown in Fig. 1 should be considered. In Fig. 1a the outflow temperature T_t^{out} in timeslot t is a mix of masses with T_{t-2}^{in} and T_{t-3}^{in} and hence must be between T_{t-2}^{in} and T_{t-3}^{in}. Thus, (7) is an outer approximation of the outflow temperature at time t.

$$\min\left(T_{t-2}^{in}, T_{t-3}^{in}\right) \leq T_t^{out} \leq \max\left(T_{t-2}^{in}, T_{t-3}^{in}\right) \tag{7}$$

To refine this outer approximation the volume leaving in timeslot t can be divided into equal sub-volumes. The sub-volume j1 (left) in Fig. 1b is at T_{t-2}^{in}. The sub-volume j2 (right) is (as above) a mix of T_{t-2}^{in} and T_{t-3}^{in} and thus must be between T_{t-2}^{in} and T_{t-3}^{in}. Hence the outer approximation of the outflow temperature becomes (8).

$$\frac{1}{2}T_{t-2}^{in} + \frac{1}{2}\cdot\min\left(T_{t-2}^{in}, T_{t-3}^{in}\right) \leq T_t^{out} \leq \frac{1}{2}T_{t-2}^{in} + \frac{1}{2}\cdot\max\left(T_{t-2}^{in}, T_{t-3}^{in}\right) \tag{8}$$

This concept can be formulated in general as MILP using the following equations.

The distance traveled Δx_t in one time slot t with duration Δt is calculated using diameter of pipe d, density of medium ρ and current mass flow $\dot{m}_t$ (9). Its initial value is length of pipe L. The discrete time delay τ is expressed as binaries $b_{\tau,j,t}$ with j as index of the sub-volume (10, 11). $\sum_{\tau=0}^{t} b_{\tau,j,t} = 1$ for all n_p sub-volumes j at all time slots t.

$$\Delta x_t = \Delta t \cdot v = \frac{\Delta t}{\rho\pi\,(d/2)^2}\dot{m}_t \tag{9}$$

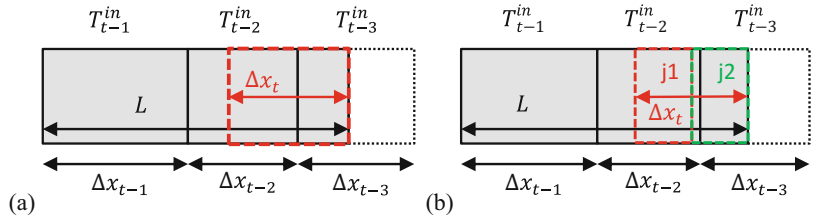


Fig. 1 Temperature distribution in a pipe of length L at beginning of timeslot t. (**a**) Volume leaving pipe in t (bold-dash). (**b**) Volume leaving pipe in t divided into sub-volumes (j1 and j2)

$$L - \frac{j-1}{n_p}\Delta x_t \leq \sum_{tt=t-\tau}^{t-1} \Delta x_{tt} + \left(1 - b_{\tau,j,t}\right) \cdot M \tag{10}$$

$$L - \frac{j-1}{n_p}\Delta x_t \geq \sum_{tt=t-\tau+1}^{t-1} \Delta x_{tt} - \left(1 - b_{\tau,j,t}\right) \cdot M \tag{11}$$

For a correct outer approximation of the temperature of a sub-volume $T_{t,j}^{out}$ the number of different possible temperatures in the sub-volume needs to be considered (12)–(14). As the number of sub-volumes is increased, the number of possible temperatures per volume decreases and hence it is sufficient to implement only the first few constraints.

$$T_{t,j}^{out} \leq T_{t-\tau}^{in} + \left(2 - b_{\tau,j,t} - b_{\tau,j+1,t}\right) \cdot T^{max} \tag{12}$$

$$T_{t,j}^{out} \leq \max\left(T_{t-\tau}^{in}, T_{t-\tau+1}^{in}\right) + \left(2 - b_{\tau,j,t} - b_{\tau-1,j+1,t}\right) \cdot T^{max} \tag{13}$$

$$T_{t,j}^{out} \leq \max\left(T_{t-\tau}^{in}, T_{t-\tau+1}^{in}, T_{t-\tau+2}^{in}\right) + \left(2 - b_{\tau,j,t} - b_{\tau-2,j+1,t}\right) \cdot T^{max} \tag{14}$$

$$T_t^{out} \leq \frac{1}{n_p} \sum_{j=1}^{n_p} T_{t,j}^{out} \tag{15}$$

(15) combines the maximum temperatures of the n_p sub-volumes to the maximum average outflow temperature of one time slot. An implementation of the lower temperature bound is not needed as the optimizer tries to maximize the temperature. Heat losses can be considered by replacing $T_{t-\tau+x}^{in}$ in (12)–(14) with (3).

(9)–(15) is a lower bound formulation of the outflow temperature. An increase of n_p increases its precision and hence the tightness of the lower bound. Thus a global optimal solution can be found using the iterative scheme of multiparametric disaggregation (see 3.1), if the lower bound problem in step 1 is extended with (9)–(15) and in step 4a (Gap too big) an increase of n_p is added.

4 Case Study

The global optimization scheme presented in Sect. 3 is used to optimize a small district heating grid consisting of two CHPs and one load area for minimum operational cost. CHP 1 has a maximum output of 400 MW heat and 500 MW electric power and a variable power-to-heat-ratio (extraction condensing turbine). It feeds the load via a pipe of length 10 km and diameter 0.7 m. CHP 2 is directly connected to the load having a maximum output of 20 MW heat and 20 MW electric

power, a fixed power-to-heat-ratio as well as a constant outflow temperature of 110 °C.

There is an electric demand which needs to be fulfilled either by power generation of the CHPs or by purchases at the day-ahead electricity market. Excess of electric energy is sold at the day-ahead electricity market (prices taken from EPEX SPOT DE/AT for November 15th, 2017). The heating demand is a typical heat demand for a November day with outdoor air temperatures from 5–10 °C. The global optimization for one day with 1 h time step is run on an Intel® Xeon® CPU E5-2660 v4 with 2 GHz, 16 cores and 16 GB RAM. Gurobi 8.0 [12] is used as MILP solver for the LB problem (time limit 1 h) and IPOPT [13] is used as non-linear solver for the UB problem.

In Figs. 2 and 3 results of the optimization are shown. At times before an increase of electricity prices the temperature in the pipe is increased and hence thermal energy is stored. This stored energy is used to supply heat demand at higher price times (hours 8–10, 12, 18–19) such that the CHPs can produce more electricity sold to the market. The solution having a gap of 8.2% was achieved in 7.392 s and 4 iterations. Smaller gaps can be achieved with increased solution times (e.g. 7.6% for MILP time limit of 10 h).

In comparison to a heat driven operation (fixed T^{out} for both CHPs) savings of about 2.3% (>7.000 €/day) could be identified. In this calculation the volume of the 10 km pipe was considered. If a higher grid volume is considered, savings will be higher.

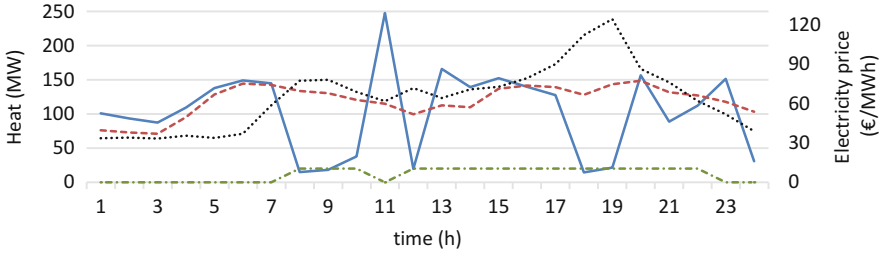


Fig. 2 Heat generation (CHP1: solid, CHP2: point-dash), demand (dash) and electricity price (dot)

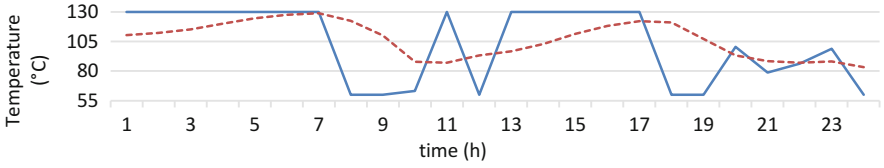


Fig. 3 Supply temperatures close to CHP 1 (solid) and at heat demand (dash)

5 Conclusion

In this paper multiparametric delay modelling is introduced as new global optimization method for heating grids and its functionality is shown for a small radial grid. Extensions to meshed heating grids with changing flow directions are possible within the concept and could be part of future developments. A drawback of the current approach is the high computational demand, which impedes a direct implementation for real world operations. Even if efforts to speed up the method are unsuccessful, it allows to benchmark existing and future (non-global) optimization schemes for optimal unit commitment and dispatch of CHPs considering the thermal inertia of district heating grids.

Acknowledgements The authors gratefully acknowledge funding by the German Federal Ministry of Education and Research (BMBF) within the Kopernikus Project ENSURE 'New ENergy grid StructURes for the German Energiewende'.

References

1. Lund, H., Østergaard, P.A., Connolly, D., Ridjan, I., Mathiesen, B.V., Hvelplund, F., Thellufsen, J.Z., Sorknæs, P.: Energy storage and smart energy systems. Int. J. Sustain. Energy Plann. Manag. **11**, 3–14 (2016)
2. Li, P., Wang, H., Lv, Q., Li, W.: Combined heat and power dispatch considering heat storage of both buildings and pipelines in district heating system for wind power integration. Energies. **10**, 893 (2017)
3. Sandou, G., Font, S., Tebbani, S., Hiret, A., Mondon, C.: Predictive control of a complex district heating network. In: Proceedings of the 44th IEEE Conference on Decision and Control and the European Control Conference, Seville, Spain (2005)
4. Loukarakis, E., Mancarella, P.: A sequential programming method for multi-energy districts optimal power flow. In: 2017 IEEE Manchester PowerTech, Manchester, UK (2017)
5. Giraud, L.: Modélisation dynamique et gestion avancée de réseaux de chaleur. PhD, Université Grenoble Alpes (2016)
6. Groß, S.: Untersuchung der Speicherfähigkeit von Fernwärmenetzen und deren Auswirkungen auf die Einsatzplanung von Wärmeerzeugern. PhD, TU Dresden (2012)
7. Li, Z., Wu, W., Shahidehpour, M., Wang, J., Zhang, B.: Combined heat and power dispatch considering pipeline energy storage of district heating network. IEEE Trans. Sustain. Energy. **7**, 12–22 (2016)
8. Schweiger, G., Larsson, P.-O., Magnusson, F., Lauenburg, P., Velut, S.: District heating and cooling systems—framework for Modelica-based simulation and dynamic optimization. Energy. **137**, 566–578 (2017)
9. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I. Convex underestimating problems. Math. Program. **10**, 147–175 (1976)
10. Teles, J.P., Castro, P.M., Matos, H.A.: Global optimization of water networks design using multiparametric disaggregation. Comput. Chem. Eng. **40**, 132–147 (2012)
11. Castro, P.M., Teles, J.P.: Comparison of global optimization algorithms for the design of water-using networks. Comput. Chem. Eng. **52**, 249–261 (2013)
12. Gurobi Optimization.: Gurobi Optimizer Reference Manual. http://www.gurobi.com (2016)
13. Wächter, A., Biegler, L.T.: On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. **106**, 25–57 (2006)

An Exact Method for Cost-Minimal Deployment of a Rail Crane Using Demand Response

Erik Pohl, Lars-Peter Lauven, and Jutta Geldermann

1 Introduction

Environmental protection and energy efficiency have become major topics in seaport management. Serving as an interface of sea and land-side, container terminals play an important role in global supply chains. On the one hand, the operators of container terminals are facing pressure to reach a higher level of throughput by shipping companies, while on the other hand, port-authorities and governments ask for energy saving and emission reduction. Therefore, many container terminals today strive to reduce emissions and energy consumption while maintaining or improving the current service level. A significant reduction of emissions in a container terminal can be achieved by using electrically-powered container handling equipment and renewable energy. The possible future electrification of equipment in container terminals will increase their power demand significantly and increase the benefit of participating in "Demand Side Management" to modify their electricity demand profile.

The overall goal is to evaluate the cost-reduction potential and grid effects of the model in container terminals. In this paper, we focus on the rail crane activities. We present a MILP to determine an electricity cost minimal schedule for a rail crane, i.e. the starting times for each container loading or unloading job of the crane with minimal costs considering volatile electricity prices. The remainder of this paper is structured as follows. Section 2 gives a brief introduction to demand response

E. Pohl (✉) · J. Geldermann
Universität Duisburg-Essen, Chair of Business Administration and Production Management, Duisburg, Germany
e-mail: erik.pohl@uni-due.de; jutta.geldermann@uni-due.de

L.-P. Lauven
Georg-August-Universität Göttingen, Chair of Production and Logistics, Göttingen, Germany
e-mail: Lars.Lauven@wiwi.uni-goettingen.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_24

and rail crane operation. In Sect. 3 we present the mathematical program and Sect. 4 shows computational results. At the end, we draw some conclusions concerning the computational complexity in Sect. 5.

2 Demand Response and Rail Crane Operation

The term "Demand Side Management" covers all actions which target to adjust of consumers demand profile in shape or time, to make it comply with the supply, while aiming at an efficient incorporation of renewable energy resources [1]. At the moment, one of the main Demand Side Management activities is demand response. The term "demand response" can be defined as "a tariff or program established to motivate changes in electric use by end-use costumers, in response to changes in the price of electricity over time, or to give incentive payments designed to induce lower electricity use at times of high market prices or when grid reliability is jeopardized" [7]. demand response therefore aims at modifying consumer demands to reduce energy demand during peak-hours to benefit the power supply system as well as reducing energy costs for the operator. Demand response programs have been extensively studied in the literature. A survey on demand response programs can be found in [8]. In the following, we analyze an electric rail crane system in a container terminal, considering volatile electricity prices to be given in 15 min periods over the whole planning horizon.

A rail crane in a container terminal is a crane to load or unload containers from trucks to trains. It is usually itself mounted onto a rail system, powered fully electric and often entirely automatic. A rail crane spans between 3 and 10 tracks. Therefore, it can handle up to 10 trains at a time. At the side of the crane system, there are spots to store containers. Depending on the type of horizontal transport system of the terminal, there are either parking spots for vehicle onto which the containers are loaded or the containers are put on the ground and later picked up by a transport vehicle (i.e. a straddle carrier). For an in-depth description of the container terminal operation and corresponding optimization problems, see [5].

3 Problem Description and Demand Response Model

This section introduces the notation and then formulates the problem as a mixed integer linear program. The crane movement in our approach is based on the model for yard cranes in [4]. In our model, the idea is to determine the order in which the crane executes given jobs using decision variables. The energy costs and starting time of each jobs leads to additional constraints using the order of execution. In the following, we introduce what we will call a job and then define the required energy and execution time for these jobs.

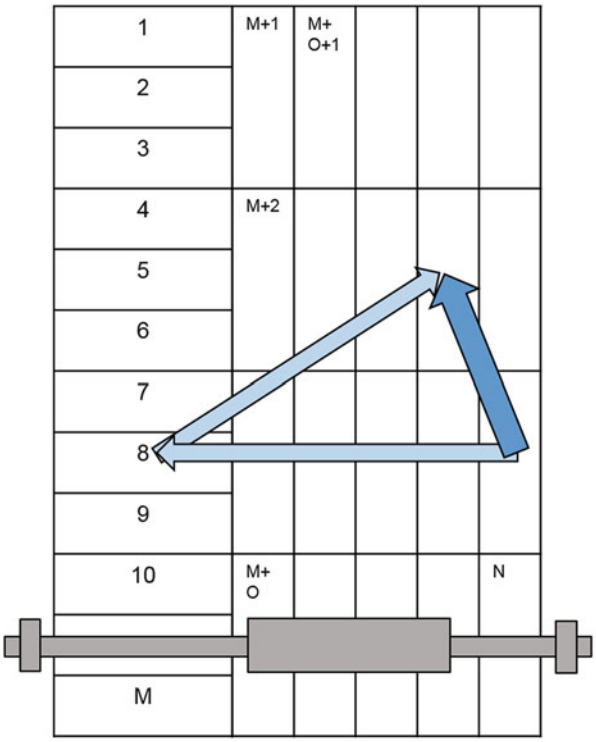


Fig. 1 Schematic view onto a rail crane system. On the left side are I/O points to put or to get container from (numbered from 1 to M), on the right side are trains onto which the container should be loaded (numbered from $M+1$ to N)

We consider a rail crane system as described in the previous section consisting of M I/O locations $I/O = \{1, \ldots, M\}$ and $N - M$ locations on trains $V = \{M+1, \ldots, N\}$. Figure 1 shows a schematic rail crane system. Furthermore, let $Z = \{1, \ldots, z^{max}\}$ be the set of trains with arrival and departure times $[a_z, d_z]$ respectively. We denote by $J^0 = \{1, \ldots, n\}$ the set of container handling jobs which should be executed by the rail crane. For each job $i \in J^0$ there is an associated start and end position as well as a time interval, in which this job must be executed. There can be two types of jobs: loading and unloading jobs. For loading jobs, the start position of a job is in I/O and the end position is V and for unloading jobs the other way around. In order to mathematically formulate the problem and determine the electricity demand and time of each job to obtain the electricity costs, we need the travel times and energy demand for all jobs. For this, let t_{ij} be the travel time for the rail crane to execute job $i \in J$ and travel to the starting position of job $j \in J$. Analogously, let e_{ij} be the required energy to execute job i and move to the starting position of job j. To model the crane movement, we need to consider designated start and end positions. For this, let $J = J^0 \cup \{s, e\}$, where s and e are

start and end points of the crane movement, respectively. We can define the travel time and required energy for these two points analogously to the above. Notice that all locations as well as destinations for each container are known. In practice, they are determined at a higher planning level and, in our model, we focus on on-time and cost minimal handling of those containers. Furthermore, to model the volatile energy prices, let $P = \{1, \ldots, p^{max}\}$ the set of 15 min time periods with associated prices $c_p \ \forall p \in P$. As mentioned before, we assume that all intervals have the same length of 15 min.

In the following we introduce the variables. To formulate the crane movement, we define the following decision variables to decide the job order. Let

$$x_{ij} = \begin{cases} 1 \text{ if job } j \in J \text{ is executed after job } i \in J \\ 0 \text{ else.} \end{cases}$$

The light-colored arrows in Fig. 1 show the actual crane movement for an example container whereas the dark-colored arrow show the movement seen from the model. The starting time of a job is denoted by the continuous variable $ST_i \ \forall i \in J$. To determine the time period $p \in P$ in which the starting time of a job $i \in J$ lies, we define the following decision variable. Let

$$z_{ip} = \begin{cases} 1 \text{ if } ST_i \text{ lies in time period } p \\ 0 \text{ else.} \end{cases}$$

The required energy of a job is denoted by $RE_i \ \forall i \in J$ and the total energy demand in a time period $p \in P$ is determined by the continuous variable $EC_{ip} \ \forall i \in J, p \in P$.

With this, we obtain the following MILP.

$$\min \quad \sum_{p \in P} \sum_{i \in J} c_p EC_{ip} \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in J \setminus \{j,e\}} x_{ij} = 1 \qquad \forall j \in J \setminus \{s\} \tag{2}$$

$$\sum_{j \in J \setminus \{i,s\}} x_{ij} = 1 \qquad \forall i \in J \setminus \{e\} \tag{3}$$

$$ST_j \geq ST_i + x_{ij} t_{ij} - (1 - x_{ij}) M \qquad \forall i, j \in J \tag{4}$$

$$a_i \leq ST_i \leq d_i \qquad \forall i \in J \tag{5}$$

$$RE_i = \sum_{j \in J} t_{ij} x_{ij} \qquad \forall i \in J \tag{6}$$

$$\sum_{p \in P} z_{ip} = 1 \qquad \forall i \in J \tag{7}$$

$$\sum_{p \in P} 15pz_{ip} \leq ST_i \leq \sum_{p \in P} 15(p+1)z_{ip} \qquad \forall i \in J \tag{8}$$

$$EC_{ip} \leq Mz_{ip} \qquad \forall i \in J, \forall p \in P \tag{9}$$

$$EC_{ip} \leq RE_i \qquad \forall i \in J, \forall p \in P \tag{10}$$

$$EC_{ip} \geq RE_i - (1 - z_{ip})M \qquad \forall i \in J, \forall p \in P \tag{11}$$

$$EC_{ip} \geq 0 \qquad \forall i \in J, \forall p \in P \tag{12}$$

$$RE_i \geq 0 \qquad \forall i \in J \tag{13}$$

$$ST_i \geq 0 \qquad \forall i \in J \tag{14}$$

$$z_{ip} \in \{0, 1\} \qquad \forall i \in J, \forall p \in P \tag{15}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in J \tag{16}$$

The objective function (1) minimizes the total energy costs. Constraints (2) and (3) model the crane movement, i.e. determine the order of jobs. We ensure that each job in J^0 has exactly one predecessor and one successor. Constraint (4) ensures that a job $j \in J$ starts at least t_{ij} minutes after its predecessor job $i \in J$. We use the commonly known $bigM$ method to disable this constraint for all non-consecutive jobs. Constraint (5) restricts the starting time of a job to its given time interval. To calculate the electricity costs which are minimized in the objective function, we need to determine the energy demand of each job and multiply with the electricity prices at the moment of execution. The energy demand of a job $i \in J$ is determined in Constraint (6). Since the starting time can only lie in exactly one period, Constraints (7) and (8) are used to link the variables z_{ip} with the starting time ST_i. The electricity demand

$$EC_{ip} = RE_i z_{ip} \quad \forall i \in J, \forall p \in P$$

is the product of a binary and a continuous variable. The latter is bounded by the maximum energy demand of a container job. Therefore, Constraints (9) to (11) are used to model this multiplication as linear constraints. Finally, Constraints (12) to (16) restrict the domains of the variables.

4 Results

In this section, we compare the computation time for different input sizes. The above model has been implemented in Gurobi 7.5 and Python 3.6 as the programming language. The program is executed on a 3.3 GHz Intel™ i5-4590 CPU with 8 GB of RAM under Windows 7. Although real-world data is available from the IHATEC

Table 1 Computational results for different input instances

#Container	#Periods	Time [s]	Objective value	Continuous variables	Binary variables	# Constraints
4	25	1.17	724	156	174	330
5	25	6.27	1237	182	210	621
6	25	43.65	1289	208	248	726
7	25	176.84	1754	234	288	835
8	25	1050.76	1970	260	330	948
4	50	0.5	206	306	324	964
5	50	3.41	450	357	385	1139
6	50	110.56	577	408	448	1318
7	50	7573.6	973	459	513	1501

project SuStEnergyPort [6], first of all the model should be tested with manageable problem sizes. The chosen test instances are randomly generated and differ in number of containers and time periods. In all instances, the travel time t_{ij} and energy e_{ij} is proportional to the distance traveled by the crane. The results presented in Table 1 confirm the complexity of the problem, as Gurobi can only solve small instances in reasonable time. Table 1 shows that the computation time increases exponentially with the number of time periods $|P|$ and the number ob jobs $|J|$. For example, planning 7 jobs in 50 time periods of 15 min takes the Gurobi-solver more than 7000 s to solve.

5 Conclusion

This paper presents a planning model for demand response of a rail crane and computational results for small instances. The objective is not to minimize the make span of the crane but to minimize the electricity costs during crane operation. In this paper, we consider one rail crane to execute all jobs. When regarding two cranes, the traveling salesman Constraints (2) and (3) would become vehicle routing constraints with exactly two vehicles. Additionally, we would need constraints to prevent the cranes from crossing over each other.

Exact models to identify the cost-saving potential of demand response in crane operation can only solve small instances. Heuristics for Traveling Salesman Problems [2] or heuristics for quay crane scheduling [3] could be used as a starting point, but need to be enhanced with time intervals and a time-dependent cost function.

References

1. Alizadeh, M., Li, X., Wang, Z., Scaglione, A., Melton, R.: Demand-side management in the smart grid: information processing for the power switch. IEEE Signal Process. Mag. **29**(5), 55–67 (2012). https://doi.org/10.1109/MSP.2012.2192951
2. Bektas, T.: The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega **34**(3), 209–219 (2006). https://doi.org/10.1016/j.omega.2004.10.004
3. Bierwirth, C., Meisel, F.: A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. Eur. J. Oper. Res. **244**(3), 675–689 (2015)
4. Gharehgozli, A.H., Yu, Y., de Koster, R., Udding, J.T.: An exact method for scheduling a yard crane. Eur. J. Oper. Res. **235**(2), 431–447 (2014). https://doi.org/10.1016/j.ejor.2013.09.038
5. Steenken, D., Voß, S., Stahlbock, R.: Container terminal operation and operations research - a classification and literature review. OR Spectr. **26**(1), 3–49 (2004). https://doi.org/10.1007/s00291-003-0157-z
6. SuStEnergyPort - Simulationsbasierte Bewertung von Maßnahmen zur Steigerung der Energienachhaltigkeit im Hafenbetrieb (2017). https://www.innovativehafentechnologien.de/wp-content/uploads/2017/08/Projektsteckbrief_SustEnergyPort.pdf
7. US Department of Energy: Benefits of demand response in electricity markets and recommendations for achieving them: a report to the united states congress pursuant to section 1252 of the energy policy act of 2005 (2006)
8. Vardakas, J.S., Zorba, N., Verikoukis, C.V.: A survey on demand response programs in smart grids: pricing methods and optimization algorithms. IEEE Commun. Surv. Tutorials **17**(1), 152–178 (2015). https://doi.org/10.1109/COMST.2014.2341586

Entity-Oriented Multi-Level Energy System Optimization Modeling

Fabian Scheller and Thomas Bruckner

1 Introductory Remarks

Energy systems are under continual change in many countries. Many of these changes take place on the municipal level. Major drivers can be seen in the rising distributed energy resources, the increasing electrification, a movement towards a carbon-neutral energy mix, changing utility business models as well as an increasing customer engagement. These trends result in a changing market environment. Decision makers need to investigate under what conditions will interventions lead to sustainable future systems. The development of strategies is a challenging task for the involved decision makers which need to consider the current business portfolio, technological progress, actor base, the regulatory framework as well as the market status.

In doing this, existing relationships between provision and utilization of energy services are necessary for the representation of actor-related social and economic activities along the energy chain. Energy policy formation over a certain municipal area is a complicated exercise involving several private and system issues and interests. Commercial actors along the energy value chain might assess challenges and opportunities from different actor perspectives and various interest criteria, since every single consumer and operator has a different technology-mediated relationship [2]. Individual households, neighborhood communities, organizational units, and market institutions represent major system drivers [8] so that an integrated view of supply and demand side is necessary for optimal investigation of the interaction at the municipal level [4]. Usually, not one single actor contains all relevant components for distribution, conversion, and the storing of energy, but

F. Scheller (✉) · T. Bruckner
Institute for Infrastructure and Resources Management (IIRM), Leipzig University, Leipzig, Germany
e-mail: scheller@wifa.uni-leipzig.de; bruckner@wifa.uni-leipzig.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_25

rather individual actors are spatially distributed over a municipal area and are equipped with respective components. Existing energy system optimization models ignore the roles different actors play in existing system architecture and the resulting impact they might have [6]. On the one hand, actors which can be prosumer groups, business units and public institutions need to be modeled separately and so do also the spatially distributed technological systems. On the other hand, multi-party cooperation needs to be incorporated. Individual actors hold bilateral contracts with each other that handle the business transactions. Thus, a comprehensive coordination design needs to merge the technical and commercial aspects and facilitate the allocations when costs and benefits do not amount to the same actor.

This research paper develops a novel, entity-oriented, multi-level optimization framework called *Integrated Resource Planning and Optimization (IRPopt)*. The modular objective function exhibits a novel formal interface between the supply and demand side merging technical and commercial aspects. This is achieved by the explicit modeling of municipal market actors on one layer and state-of-the-art technology processes on another layer as well as resource flow interrelations and service agreements mechanism among and between the different layers.

The research paper is organized as follows: Sect. 2 introduces a previously established graph-theoretical approach to the technical energy system. Building on this, Sect. 3 proposes a novel entity related coordination modeling approach. The integration of both layers in terms of a multi-level optimization framework is presented in Sect. 4. Finally, Sect. 5 summarizes the main contributions and outlines future work.

2 Energy Process Graph Modeling

An energy process graph represents the technical foundation of the framework. This approach allows a flexible configuration of sectors, processes, and the interconnections. Similar graph implementations have been applied in established optimization models like *deeco* [1], *EnergyHub* [5] and *urbs* [3].

While the *engineering processes* of the system (e.g. transmission, storage, conversion, collector and demand) are denoted by $n \in N$, the *energy sectors* of the system (e.g. electricity, heating and gas) are given by $u \in U$. The emerging *energy system graph* $E = (N, A)$ resembles a *multi-commodity flow graph* with the set of engineering processes N as nodes and the set of *energy links* A as arcs ($A \subset U \times N \times N$). Arcs as links are represented by $a = (u, n, n')$ as ordered 3-tuple with source node $n \in N$, target node $n' \in N$, and sector index $u \in U$. Every directed arc of the multi-commodity flow and store network is associated with exactly one energy carrier and thus can be distinguished by the sector. The arcs between the same nodes can be distinguished by the sector $u \in U$. By taking into account that flows are naturally directional, a directed arc $a \in A$ is defined as follows $a = (u, n, n'), a' = (u', n, n') \Rightarrow (u \neq u') \vee (a = a')$. An example is illustrated in Fig. 1.

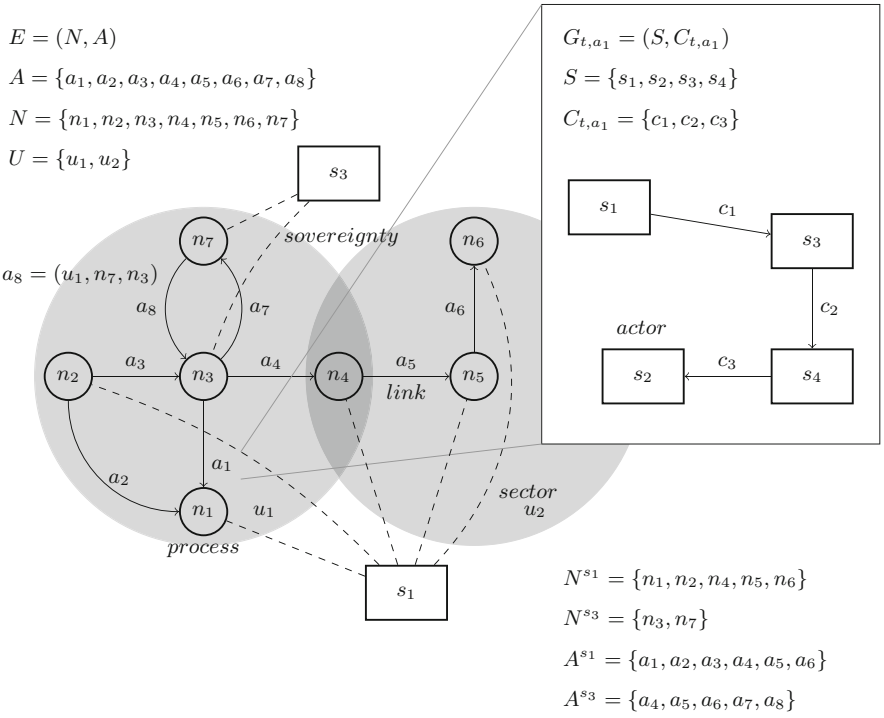


Fig. 1 Schematic representation of an exemplary configuration of the energy process graph $E = (N, A)$ and a commercial entity coordination graph $G_{t,a} = (S, C_{t,a})$

The energy flow in [MWh] through each arc $a \in A$ is described through the *endogenously determined energy flow function* $\mathfrak{e} : T \times A \to \mathbb{R}_{\geq 0}$ representing the flow of energy through arc $a \in A$ at time $t \in T$. This optimal operation policy is dependent on monetary flows which are exchanged between commercial actors.

3 Commercial Entity Coordination Modeling

The entity coordination facilitates the commercial interactions between market actors and thus represents the economic foundation of the framework. The underlying idea of commercial relationships is inspired by the descriptions of authors of the optimization model *xeona* [7]. In this context, the modeling of individual entities or rather *commercial actors* forms the basis for the integrated multi-level analysis. A specific actor is denominated by $s \in S$. Relevant actors might be prosumer groups ($S^{pros} \subset S$), utility business units ($S^{orga} \subset S$) or public institutions ($S^{inst} \subset S$). Each one represents a possible *contractual partner* regarding various commercial interactions.

Not one single actor contains all relevant engineering processes for distribution, conversion, and storing of energy, instead, individual actors are spatially distributed over a municipal area and are equipped with respective processes. Thus, a function regarding the ownership of the processes is required. This *sovereignty function* $\mathfrak{n} : S \rightarrow Pot(N)$ indicates which actor has the authority to operate the process n, meaning that the variables associated with it are free to dispatch variables. $Pot(N)$ describes the power set of any set N. Considering the possession of the individual processes $n \in N$, the specific actor $s \in S$ may only operate the energy flow along arcs adjacent to a node n they have optimization sovereignty over ($n \in N^s$ with $N^s = \mathfrak{n}(s) \subset N$). Thus, the set of energy links an actor can operate during the optimization run might be determined as follows: $A^s = \{a = (u, n, n') \in A : n \in N^s \vee n' \in N^s\}$.

Regarding the linkage of the actors and the processes energy flows in E as outlined in Sect. 2 need to be interrelated with the financial flow between actors. This is done by associating each energy flow (through arc $a \in A$ in the energy system graph at time $t \in T$) with a set of *exogenously given energy tariffs*. Thus, actors are able to optimize the energy flow from an economic perspective.

Coordination relationships between actors for these energy flows are depicted through energy tariff graphs. Since a multitude of contractual relationships can exist for a given energy flow (e.g. the consumer pays the sales department for the consumed energy, which in turn pays the network and the trading unit for the provision and energy supply). These can differ in their dependence of the flow of energy and the point in time (e.g. fixed energy tariffs over time or flexible energy tariffs over time depending on the amount of energy dispatched) energy tariffs are formalized through a set of energy tariff graphs $G = \bigcup_{a \in A} \bigcup_{t \in T} G_{t,a}$. An illustration is outlined in Fig. 1.

Each *exogenous tariff graph* $G_{t,a}$ is a directed graph $G_{t,a} = (S, C_{t,a})$ describing the contractual relationships between actors $s \in S$ for the given energy flow along arc $a \in A$ in the energy system graph E at time $t \in T$ with $C_{t,a} \subset S \times S$. Each energy tariff graph arc $c \in C_{t,a}$ is associated with a corresponding tariff in [/MWh] via the energy tariff function $\mathfrak{w} : T \times A \times S \times S \rightarrow \mathbb{R}_{\geq 0}$. The function specifies the contractual relationship between the actors $s \in S$ for the energy flow from component $n \in N$ to $n' \in N$ at a time $t \in T$ for a given sector $u \in U$.

The presented division of the technical and economic graphs on different layers allow a flexible mapping of the actual conditions. This also delimit IRPopt from existing system models. While inter alias *deeco* only considers financial flows at the system boundaries and the whole domain [1], IRPopt dissolves the domain and equips various tariffs for subdomains. Thereby, actors can also be incorporated even though they are not directly tangent.

4 Multi-Level Optimization Framework

4.1 Objective Function

IRPopt represents a bottom-up techno-economic optimization model, implemented in GAMS/CPLEX, for solving mixed-integer problems (MIP). The mathematical objective function of IRPopt is based on the determination of the optimal dispatch of the designed energy process graph considering the actor related coordination graphs. The major objective is to maximize revenues (maximize $\mathfrak{f}$) as expressed in Eq. (1). For completion, $\mathfrak{f}$ is restricted by different equality constraints $\mathfrak{g} \in \{\mathbb{R}_{\geq 0}, \mathbb{Z}_{\geq 0}\}$ and inequality constraints $\mathfrak{h} \in \{\mathbb{R}_{\geq 0}, \mathbb{Z}_{\geq 0}\}$ which describe the technical nature of the engineering processes.

$$\text{maximize} \quad \mathfrak{f} = \sum_{t \in T'} \sum_{s \in S'} \sum_{a \in A^s} \left(\sum_{s' \in S} \mathfrak{e}_{t,a} \cdot \mathfrak{w}_{t,a,s',s} - \sum_{s' \in S} \mathfrak{e}_{t,a} \cdot \mathfrak{w}_{t,a,s,s'} \right) \tag{1}$$

For entity-oriented multi-level optimization, the set of time points of interest $T' \subset T$ as well as the set of actors of interest $S' \subset S$ need to be specified to define the optimization period and perspective. According to the design decision taken, associated energy links ($\bigcup_{s \in S'} A^s \subset A$ of the actor $s \in S'$) are determined. The optimal operation is calculated by maximizing the profits from the chosen actor perspective. In this context, the energy flows $\mathfrak{e}$ associated with the actor(s) of interest are dispatched on the basis of the energy tariffs $\mathfrak{w}$ associated with it.

4.2 Application Procedure

For application, it is particularly suitable to subdivide the time-steps of the scheduling horizon (e.g. representative year of hourly quarter-hourly time steps, $T' \subset T = \{1, \ldots, 35040\} \subset \mathbb{N}$) to enable a recursive dynamical optimization. A major reason is that the optimization of only time-local problems might reduce computational complexity. Additionally, considering the exemplary configuration of Fig. 1, an entity-oriented, two-step optimization procedure is conceivable. Initially, the perspective or rather set of actors of interest might be defined with $S' = \{s_1\}$. This allows to determine all processes and links the defined set of actor has regulative access to on the basis of the functions N^{s_1} and A^{s_1} as well as to calculate the optimal energy and financially related dispatch. Subsequently, the energy and financial flows of the exemplary model configuration might be optimized from the second perspective with $S' = \{s_3\}$. Thereby, the residual energy demand and supply of the previous dispatch is considered.

By default, IRPopt initially optimizes from an aggregated prosumer' perspective ($S^{'} = S^{pros}$), determining the residual energy demand and excess energy supply with all processes the customers have regulative access to. Subsequently, the model optimizes all other energy and financial flows from the utilities' perspective ($S^{'} = S^{orga}$). Such an idea of a multi-level entity-oriented optimization approach has already been initially outlined by Morrison et al. [7]. The applications published in [9, 10] demonstrate the modeling capabilities of the developed optimization framework by assessing demand response and community storage from different actor perspectives. A web-based software environment supports the utilization of the decision-model by providing a scenario and data management system.

5 Concluding Remarks

Energy system strategy decisions ought to be supported by optimization studies. This research paper presents an entity coordination approach which is directly based on a proven energy system graph approach. IRPopt is able to determine optimal operation strategies from the perspective of different actors rather than a universally optimal operation strategy from the perspective of a centralized planner. With this in mind, the optimization model is suitable to assess business models that should offer an opportunity for different market actors, such as municipal utilities and residential prosumers.

References

1. Bruckner, T.: Dynamische Energie -und Emissionsoptimierung regionaler Energiesysteme. Dissertation, Universität Würzburg (1996)
2. Bruckner, T., Morrison, R., Wittmann, T.: Public policy modeling of distributed energy technologies: strategies, attributes, and challenges. Ecol. Econ. **54**(2), 328–345 (2005). https://doi.org/10.1016/j.ecolecon.2004.12.032
3. Dorfner, J.: Open source modelling and optimisation of energy infrastructure at urban scale. Dissertation, TU München (2016)
4. Farzaneh, H., Doll, C.N., Puppim de Oliveira, J.A.: An integrated supply-demand model for the optimization of energy flow in the urban system. J. Clean. Prod. **114**, 269–285 (2016). https://doi.org/10.1016/j.jclepro.2015.05.098
5. Geidl, M.: Integrated modeling and optimization of multi-carrier energy systems. Dissertation, ETH Zürich (2007)
6. Koirala, B.P., Koliou, E., Friege, J., Hakvoort, R.A., Herder, P.M.: Energetic communities for community energy: a review of key issues and trends shaping integrated community energy systems. Renew. Sustain. Energy Rev. **56**, 722–744 (2016). https://doi.org/10.1016/j.rser.2015.11.080
7. Morrison, R., Wittmann, T., Heise, J., Bruckner, T.: Policy-oriented energy system modeling with xeona. In: Proceedings of ECOS: Shaping our Future Energy Systems (2005)

8. Pfenninger, S., Hawkes, A., Keirstead, J.: Energy systems modeling for twenty-first century energy challenges. Renew. Sustain. Energy Rev. **33**, 74–86 (2014). https://doi.org/10.1016/j.rser.2014.02.003
9. Scheller, F., Burgenmeister, B., Kondziella, H., Kühne, S., Reichelt, D.G., Bruckner, T.: Towards integrated multi-modal municipal energy systems: an actor-oriented optimization approach. Appl. Energy **228**, 2009–2023 (2018). https://doi.org/10.1016/j.apenergy.2018.07.027
10. Scheller, F., Krone, J., Kühne, S., Bruckner, T.: Provoking residential demand response through variable electricity tariffs - a model-based assessment for municipal energy utilities. TESG **3**(1), 7 (2018). https://doi.org/10.1007/s40866-018-0045-x

Part VI
Finance

The Impact of Monetary Policy on Investment Bank Profitability in Unequal Economies

Bernard M. Gilroy, Alexander Golderbein, Christian Peitz, and Nico Stoeckmann

1 Introduction

Does the European Central Bank (ECB) destroy commercial banking?

The aim of this article is to demonstrate, theoretically and descriptively, that the ECB interest rate policy has a huge negative impact on the European banking sector justified by composition of firm structures within this industry. Since the financial crisis in 2007/2008, the ECB lowered its key interest rate.

Being initially an aggressive response to counteract the financial and European crisis in the European Monetary Union (EMU), the prolonged and persistent low interest rate policy of the ECB seems to have become mainstream. This policy not only intends to prevent a macroeconomic downturn and collapse but also aims at keeping the inflation rate in the EMU at a stable level.

However, a long lasting low interest rate environment entails side effects because it can be perceived as an unnatural situation for consumers expectation and leads to structural changes in the financial sector. Especially the banking sector struggles with the current situation resulting in a realignment of their strategic direction regarding income sources in order to ensure profitability. Recent studies suggest that unusually low interest rates have a negative impact on bank profitability [5].

As this result may hold for banks on average, a more detailed look at the core businesses of banks may give a different picture. As bank profitability for commercial banks who largely depend on the yield curve erodes on average, investment banks benefit from low interest rates as capital asset prices rise.

B. M. Gilroy · A. Golderbein · C. Peitz (✉) · N. Stoeckmann
University of Paderborn, Paderborn, Germany
e-mail: christian.peitz@upb.de; christian.peitz@uni-paderborn.de;
http://www.uni-paderborn.de/person/2980/

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_26

Based on a theoretical hypothesis, we observe that stock prices of investment and commercial banks react differently in the short run after the historical announcement of the ECB to lower the key interest rate to 0.00%.

Economic theory as well as broad econometric work on stock indices show the correlation between lowering the ECB key interest rate, respectively the Federal Reserve System (FED) fundamental fund rate, and boosting the economy [4, 11]. Jobst and Lin [12] argue that rate cuts outweigh the benefits from higher asset values and stronger aggregate demand. Ehrmann and Fratzscher [9] show industry specific effects of US monetary policy. But even within the banking industry we find heterogeneity in many forms [13]. There is asymmetric interest rate pass-through [2] and different outcomes on different banking system stability [7] on an international level. Ehrmann et al. [10] find international spill over effects between US and EU stock markets as a transmission of shocks. A tightened monetary policy also brings dynamics to the bank itself [8]. With quarterly balance sheet information for listed banks in the EU and USA, [1] find evidence that low interest rates contribute to banks' risk. Policy uncertainty decreases stock prices in the average [15], that is why we focus especially on this time point, where the ECB interest rate is set to zero for the first time in history.

2 Monetary Policy

Historically, the Great Depression was what may be seen as an intellectual failure of the economic policy makers of the day working on business cycle theory as macroeconomics was ordinarily called. As is well known and documented, John Maynard Keynes commanded the controversial position of the relevance of effective aggregate demand versus notional aggregate demand issues. It was argued that it is aggregate demand that determines the overall level of economic activity. Thus being, government intervention was a necessity to accommodate the observed cycles of economic activity. We focus briefly here on the so-called "new" monetary policy design attempts under ECB directed by Mario Draghi since November 1, 2011. It appears to be the case that a rethinking of policy design is needed. Since the deregulation of the financial sector in the 1980s, central banks around the world have been attracted to implementing various types of unconventional monetary policy.

Taking a look backwards at Keynes magnum opus, one observes that the classical economists belief in Say's Law, that implies that supply creates its own demand did not appear to reflect reality as the concept of price stickiness did not permit an effective interaction of aggregate demand and aggregate supply to stabilize the world's economy. As Keynesian macroeconomics emerged various building blocks of modern day macroeconomics were established as traditional knowledge:

– The relation of consumption to income and the derived multiplier effects.
– Liquidity preference aspects regarding the demand for money.

– The importance of expectations in affecting observed consumption and investment patterns and the idea that animal spirits are a major element behind shifts in demand and output.

Put in a nutshell various historical schools of thought gave rise to macroeconomic predictions seemingly based upon doctrines that are fundamentally flawed. In brief, by the end of the 1980s, the challenges raised by the rational-expectations Lucas critique [6] have led to a total overhaul of traditional macroeconomic beliefs.

During the so-called "Great Moderation" central banks began to lower interest rates in view of stabilizing the economy in order to supposedly ensure full employment in times of demand shortage and low inflation. As [14] among others are convincingly pointing out, economists seem to have forgotten Keynes message that interest rates may not solve the problem. Savings and Investment may not adequately adjust to lower interest rates, resulting in a collective disaster instead of the desired collective optimum.

The question now is, is the unconventional monetary policy of low interest rates offered by the ECB negatively influencing the future role of national commercial banks as opposed to global investment banks such that money or unconventional monetary policy may not be neutral, but may indeed have real balance effects? Do interest rate induced changes in relative prices between stocks and credit cause a reshuffling or rebalancing of banks portfolios?

3 Pricing Stocks in a Heterogeneous Bank Industry

$$P_{j0} = D_{j1}/(1+i) + D_{j2}/(1+i)^2 + \ldots + (D_{jn} + P_{jn})/(1+i)^n$$
$$= \sum_{t=1}^{n} D_{jt}/(1+i)^t + P_{jn}/(1+i)^n \quad (1)$$

The current stock price of bank j is represented by P_{j0}. Expected future Dividends $D_{j1} \ldots D_{jn}$ are discounted by i in the corresponding period t. If $lim_{n\to\infty}$, the second part of the sum in Eq. (1) vanishes and only expected dividends define the stock price P_{j0}.

$$P_{j0} = \sum_{t=1}^{\infty} D_{jt}/(1+i)^t \quad (2)$$

Now stock prices are defined by two factors D and i. We only consider the expected dividends since the interest rate is homogeneous across the industry. Of course dividend payments strongly depend on shareholder management policies within a firm. For simplicity we assume a constant firm specific d_j over time, for the part of the profits that is passed on to the shareholders. We include a wide range

of firms in the empirical section to cover the heterogeneity in d. Splitting the expected dividend payments into the two core businesses of a bank, investment and commercial banking, define D_{jt} as a linear combination times d_j.

$$D_{jt} = d_j(\lambda \Pi_{Inv_{jt}} + (1 - \lambda)\Pi_{Com_{jt}}) \tag{3}$$

λ is in the range between 0 and 1 depending on how much the core business of the bank tends to investment. If λ is equal to 1, we observe a pure investment bank. If λ is 0, it is a pure commercial bank. In reality we observe hybrids within the interval [0;1]. A general version would be $D_{jt} = f(\lambda \Pi_{Inv_{jt}}, (1 - \lambda)\Pi_{Com_{jt}}, X)$ with the dividends as a function of profits in the investment and commercial sector of the regarding bank plus a term X to cover factors, which can not be contributed to neither $\Pi_{Inv_{jt}}$ nor $\Pi_{Com_{jt}}$. For simplicity and to illustrate our hypothesis in Fig. 1 we assume linearity. In [3] words, the business of banking ought to be simple; if it is hard it is wrong.

The ECB tries to boost economy, respectively investments I, in the long run and therefore lowers the interest rate i.

$$i \downarrow \Rightarrow I \uparrow \Rightarrow \Pi_{Inv} \uparrow,\ \Pi_{Com} \downarrow \tag{4}$$

Regarding Eq. (4) a decreasing interest rate i has two effects on banks' businesses. On the one hand the investment business is boosted. On the other hand the commercial business suffers from lower interest rates. We assume that this effect can also be observed within the bank industry itself as banks are heterogeneous with $\lambda \in [0; 1]$ sketched in Fig. 1.

Lowering the ECB base rate is a general statement to boost the economy. Therefore we expect non-bank stock prices to rise, while banks with a low λ, rather commercial banks, suffer.

Figure 2 shows the performance of three leading US and EMU banks starting with the last major ECB key interest rate decision, announced on March 10, 2016 up to the end of 2016. It can be clearly seen that the performance of the EMU banks' (Deutsche Bank, Unicredit and Intesa Sanpaolo) declines during this period,

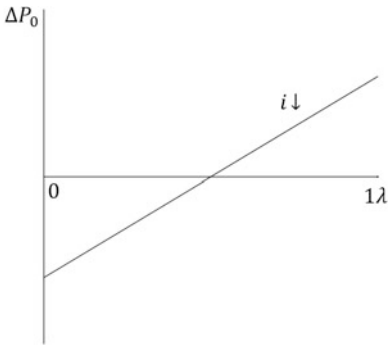


Fig. 1 Bank's stock price response depending on core business

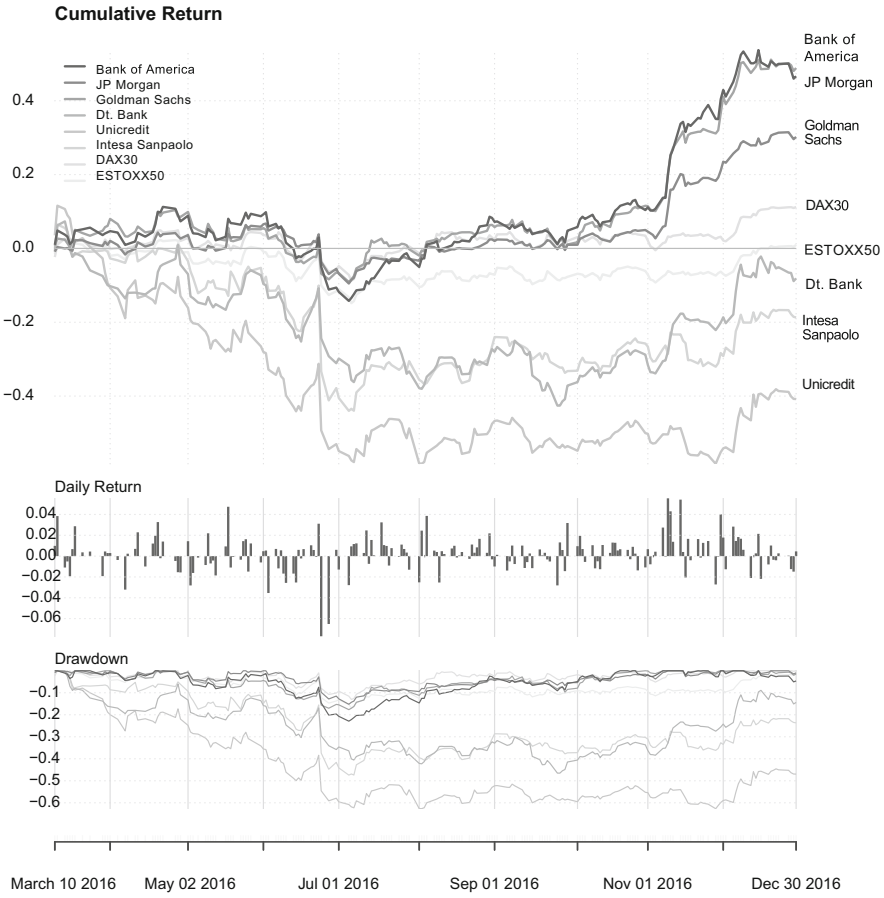


Fig. 2 Response of stock prices of EU investment banks to interest rate announcements

while the market, illustrated by DAX30 and EStoxx50, increases. The values of the three major US banks (Bank of America, JP Morgan and Goldman Sachs) show that the global banking industry has no structural problem. The break on June 24 in all time series is due to the Brexit decision. Furthermore, there is a strong increase on November 2016, especially among US banks. This is due to the election of Donald Trump on November 8, 2016. European banks are also benefiting from the network effects among markets. For the following analyses we therefore limit the period to a maximum of 6 month until September 10, 2016.

Figure 3 shows the relative Risk and Return of the respective banks. In this context, it can be seen that the European banks have a significantly higher risk with a low return. Therefore, the volatility is also increasing very strongly. As expected, the two indices are at a medium level.

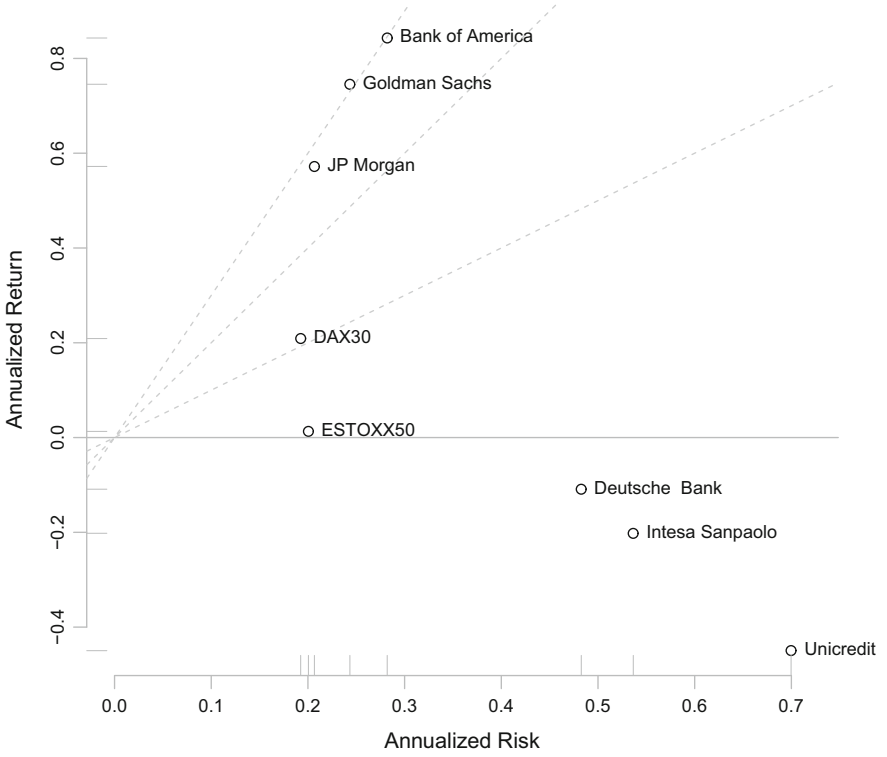


Fig. 3 Relative risk and return

It was decided at the ECB meeting on March 10, 2016 that the interest rate on the main refinancing operations of the Eurosystem would be reduced to 0.00% entering into force on March 16, 2016. Considering the short-term rates of change: On the same day, bank stocks fell significantly by 4.53% (Deutsche Bank), 0.84% (Unicredit) and 2.20% (Intesa Sanpaolo). Also on the following day the shares reacted as expected with significant losses of 2.05% (Deutsche Bank), 3.73% (Unicredit) and 4.32% (Intesa Sanpaolo). Reflecting the whole economy, corresponding indices lost less (<1% each).

For the sake of completeness, Table 1 shows the numerical changes within the considered banks. The key interest effect among other effects seems to have an impact on the failure of European banks. Until developing a valid regression model, we cannot clearly state the size of the influence.

As a supplementary, Fig. 4 shows the abnormal positive and negative returns of the Deutsche Bank compared to the corresponding market index DAX30. This clearly shows that the returns are abnormally negative, especially after the event (the decision on March 10, 2016).

Table 1 Monthly returns within the period under review

Bank/indices	After 3 weeks (10.03.16–31.03.16)	After 3 month (10.03.16–10.06.16)	After 6 month (10.03.16–10.09.16)
Bank of America	2.85%	5.12%	18.05%
JP Morgan	0.16%	7.68%	11.99%
Goldman Sachs	4.61%	−0.01%	11.73%
Deutsche bank	−14.14%	−20.45%	−23.23%
Unicredit	−12.76%	−41.78%	−42.75%
Intesa Sanpaolo	−4.51%	−19.02%	−16.67%
DAX30	2.46%	1.14%	8.38%
ESTOXX50	−0.17%	−4.86%	−4.97%

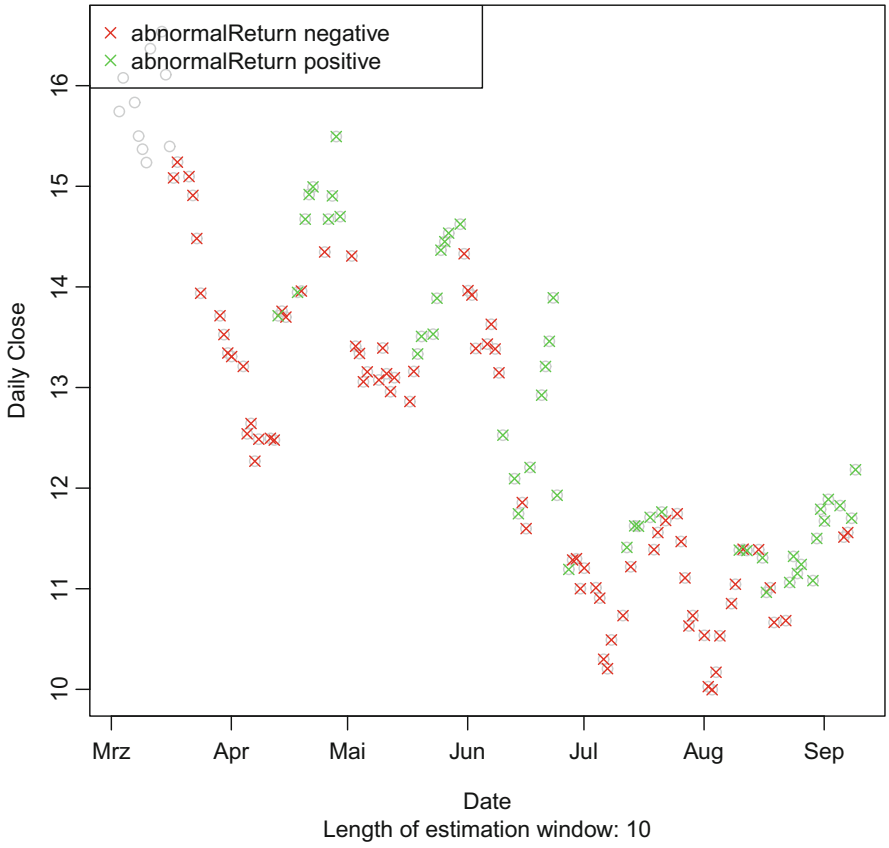


Fig. 4 Abnormal positive and negative returns

4 Conclusion

Given that the challenge of unconventional monetary policy and its effects on transmission channels of the quantity of bank credit in the finance sector and inherently the riskiness of bank portfolios as discussed above have shifted, there is an intrinsic need for macro prudential tools to safeguard future macroeconomic stability avoiding one financial disruption after another. We need to develop a more fundamental understanding of liquidity-driven unconventional monetary policy. Old ideas that have entrapped our thinking need to be replaced with new real world economic understandings.

The negative performance of European banks is based on the ECB monetary policy. A distinction between commercial and investment banks is not possible in Europe since there is no pure investment bank. We assume that banks are within the interval $\lambda \in [0; 1]$ and as they approach the investment bank limit with $\lambda = 1$ they perform better on the stock market than commercial banks.

References

1. Altunbas, Y., Gambacorta, L., Marques-Ibanez, D.: Does Monetary Policy Affect Bank Risk-Taking? (2010)
2. Apergis, N., Cooray, A.: Asymmetric interest rate pass-through in the US, the UK and Australia: new evidence from selected individual banks. J. Macroecon. **45**, 155–172 (2015)
3. Bagehot, W.: Lombard Street: A Description of the Money Market. Armstrong & Company, Scribner (1873)
4. Bernanke, B.S., Kuttner, K.N.: What explains the stock market's reaction to Federal Reserve policy?. J. Financ. **60**(3), 1221–1257 (2005)
5. Borio, C., Gambacorta, L., Hofmann, B.: The influence of monetary policy on bank profitability. Int. Financ. **20**(1), 48–63 (2017)
6. Brunner, K., Meltzer, A.: Econometric policy evaluation. A critique. In: Theory, Policy, Institutions: Papers from the Carnegie-Rochester Conferences on Public Policy, vol. 1, North Holland (1983)
7. Burriel, P., Galesi, A.: Uncovering the heterogeneous effects of ECB unconventional monetary policies across euro area countries. Eur. Econ. Rev. **101**, 210–229 (2018)
8. Den Haan, W.J., Sumner, S.W., Yamashiro, G.M.: Bank loan portfolios and the monetary transmission mechanism. J. Monet. Econ. **54**(3), 904–924 (2007)
9. Ehrmann, M., Fratzscher, M.: Taking stock: Monetary policy transmission to equity markets. J. Money Credit Bank. **36**(4), 719–737 (2004)
10. Ehrmann, M., Fratzscher, M., Rigobon, R.: Stocks, bonds, money markets and exchange rates: measuring international financial transmission. J. Appl. Econ. **26**(6), 948–974 (2011)
11. Galí, J., Gambetti, L.: The effects of monetary policy on stock market bubbles: some evidence. Am. Econ. J. Macroecon. **7**(1), 233–57 (2015)
12. Jobst, A., Lin, H.: Negative Interest Rate Policy (NIRP): Implications for Monetary Transmission and Bank Profitability in the Euro Area. International Monetary Fund, Washington (2016)
13. Kakes, J., Sturm, J.-E. Monetary policy and bank lending: evidence from German banking groups. J. Bank. Financ. **26**(11), 2077–2092 (2002)
14. Palley, T.I.: Why ZLB economics and negative interest rate policy (NIRP) are wrong: a theoretical critique. No. 172. In: IMK Working Paper (2016)
15. Pastor, L., Veronesi, P.: Uncertainty about government policy and stock prices. J. Financ. **67**(4), 1219–1264 (2012)

A Mathematical Programming Approach for the Optimal Collateral Allocation Problem

Konstantinos Papalamprou, Efthymios P. Pournaras, and Styliani Tychalaki

1 Introduction

The development of an optimal collateral allocation strategy constitutes a very important issue for financial institutions in the effort to estimate the actual risk they are exposed to. Following regulatory guidelines and rules, these institutions should reserve capital in order to be protected by expected and potential unexpected losses (estimated by means of prescribed formulas provided for by the regulators)[2]. Moreover, it is well-known that the main credit risk mitigation factor is the value and the quality of the associated set of collaterals. In the vast majority of cases, a many-to-many relationship exists between a given set of loans and the set of associated collaterals. Thus, given a set of loans associated with the same set of collaterals, utilizing the amount of the underlying collaterals efficiently has a major impact on the growth and strategy of the financial institutions in the sense that the lower the capital held because of regulatory requirements the more the capital available for running business.

Various methods to minimize the regulatory capital via optimal allocation of collaterals have been proposed in the relevant literature. In our work, three different approaches are being examined: (1) a simple method based on a rule of proportional allocation, (2) an integer linear programming formulation suggested by a researcher affiliated with a major investment bank (3) a method developed by the authors of this work which is based on a variant of the well-known transportation problem. In the context of our approach, the relationships between collaterals and loans

K. Papalamprou (✉) · S. Tychalaki
Aristotle University of Thessaloniki, Thessaloniki, Greece
e-mail: papalamprou@ece.auth.gr; tych.stella@gmail.com

E. P. Pournaras
National and Kapodistrian University of Athens, Athens, Greece
e-mail: mpournaras@econ.uoa.gr

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_27

are represented by means of a bipartite graph, namely, the set of collaterals and that of loans form the disjoint sets of vertices of that graph. An arc between two vertices denotes that a collateral may be used for risk mitigation purposes for the associated loan/exposure. The connected components of this graph are extracted based on the given collateral-loan relationships in order to reduce the time needed for analysis and the complexity of the problem. Next, the level of capital that should be held for regulatory purposes is estimated using the aforementioned approaches in each cluster separately in order to obtain the total regulatory capital needed. A synthetic dataset is being used along with a randomly selected sub-portfolio from a major Greek bank. The result of our implementations indicate that the methodology proposed in the current work provides the lowest estimations.

2 Formulations of the Collateral Allocation Problem

Usually a number of collaterals is available to cover a set of loans and, thus, a collaterals-to-loans allocation problem does exist and one of the financial institutions' objectives is to keep the regulatory capital as low as possible. In the current work we examine three different methodologies for allocating collaterals to loans. We should note that we assume that the net values of the collaterals are available, i.e. we assume that the values of the collaterals were estimated following a collateral type-based approach, which is beyond the scope of the analysis in hand. In what follows we denote by L_i the exposure/amount of loan $i \in \{1, \ldots, N\}$, by C_j the value of the collateral $j \in \{1, \ldots, M\}$ and by d_{ji} the percentage of (the value of) collateral j allocated to cover the exposure associated with loan i. It is evident that the allocation is determined by the values of the d_{ji}s and, thus, they form the set of variables in the methodologies provided below. In all cases a preprocessing step is required in order to reduce the computational complexity of the procedure. Specifically, the collaterals in hand correspond to the vertices of the one part of a bipartite graph while the loans correspond to the vertices of the other part. An undirected arc does exist in the bipartite graph if only if a collateral is associated with a loan. Then, we can use one of the various methodologies to extract the connected components of the bipartite graph (in our case we have slightly adapted the Hoshen-Kopelman algorithm [5]), which we shall call clusters, and each allocation methodology may be then applied to each cluster independently. Thus, in what follows we can assume that we work on a single cluster.

Proportional Allocation Method It is based on the seemingly logical assumption that each d_{ji} should be determined by the ratio of L_i to the total amount of loans that can be covered by the collateral j, denoted by $L^{(j)}$, i.e.: $d_{ji} = \frac{L_i}{L^{(j)}}$. The aforementioned allocation rule is simple, however, it fails to provide an optimal allocation scheme in various cases.

Cutaia Method [3] Two optimality criteria were used in the formulation presented by M. Cutaia (affiliated with Credit Suisse investment banking company in 2011):

(1) the coverage ratio of every loan of a cluster to be as close as possible to the coverage ratio of the whole cluster, i.e. for each loan i (in a cluster), minimize the $|\sum_j d_{ji} C_j / L_i - \alpha_0|$, where $\alpha_0 = \sum_j C_j / \sum_i L_i$ is the average cluster coverage ratio.
(2) the percentage allocation of each collateral j to the associated loans should be as close to $d_{j0} = 1/((\text{number of loans associated with j})$, i.e. minimize $\sum_i \sum_j |d_{ji} - d_{j0}|$.

The exact mathematical formulation of the optimization problem goes as follows:

$$min \sum_{i=1}^{N} |\sum_{j=1}^{M} d_{ji} C_j / L_i - \alpha_0| + \sum_{i=1}^{N} \sum_{j=1}^{M} |d_{ji} - d_{j0}| \tag{1}$$

subject to:

$$\sum_{i=1}^{N} d_{ji} = 1 \qquad \forall j$$

$$d_{ji} \geq 0 \qquad \forall i, j$$

$$d_{ji} = 0 \qquad \forall i, j, \text{ if } j \text{ is not connected to } i$$

where the main set of constraints ensures that all the available collateral value will be allocated to some of the associated loans. In [3], this problem is easily transformed to a linear programming problem using standard operations research techniques and, thus, it can be solved fast by optimization software suites.

Transportation Method In our formulation, we view the collaterals as source nodes that distribute parts of their value to their associated loans which correspond to destinations in our formulation based on the well-known transportation problem (see e.g. Chapter 8 in [4]). Specifically, we denote by x_{ji} the amount of collateral j which is assigned to loan i and by y_i the amount of exposure that it is not covered by any collateral available. Moreover, we make use and adapt the notion of risk weight factor (RW) which appears in the relevant credit risk guidelines and regulations [2] and transforms a given sets of exposures into risk-weighted assets. The RW depends on the type of collateral and the type of exposure/loan and, thus, it appears as RW_{ji} in our following formulation while, for the uncovered portion of a loan, it depends clearly only on the type of exposure/loan and it appears as RW_i in the formulation.

$$min \sum_{i=1}^{N} \sum_{j=1}^{M} RW_{ji} x_{ji} + \sum_{i=1}^{N} RW_i y_i \tag{2}$$

subject to:

$$\sum_{j=1}^{M} x_{ji} + y_i = L_i \qquad \forall i$$

$$\sum_{i=1}^{N} x_{ji} \leq C_j \qquad \forall j$$

$$x_{ji} = 0 \qquad \forall i, j, \text{ if } j \text{ is not connected to } i$$

$$x_{ji} \geq 0 \quad \forall i, j \quad \text{and} \quad y_i \geq 0 \quad \forall i$$

We may assume that the value of the objective function provides a good approximation of the value of the risk weighted assets (RWA) of the loan portfolio and the set of collaterals in hand, since it may be viewed as a simplified version of the formula provided in the regulatory documents [1, 2]. Moreover, regarding the risk weight factors, we have assumed that covered amounts of exposures are weighted by a factor equal to 1, i.e. $RW_{ji} = 1 \; \forall i, j$, while the uncovered ones by a factor equal to 1.5, i.e. $RW_i = 1.5 \; \forall i$.

In order to compute the RWA produced by the two other methodologies (i.e. the proportional allocation and the Cutaia methodologies) we can use the following equivalent formula:

$$RWA = \sum_{i=1}^{N} \left[\sum_{j=1}^{M} RW_{ji} C_j d_{ji} + \left(RW_i \left(L_i - \sum_{j=1}^{M} C_j d_{ji} \right) \right) \right]$$

Finally, following the standard regulatory rule appearing in [2] we have computed: Regulatorycapital $= 8\% RWA$.

3 Results

Real-World Dataset

The actual data were retrieved from an 1-month snapshot of a systemic Greek bank's corporate portfolio. The loans belong to the category of usual corporate term loan credit products. On the contrary, the collaterals cover a wide range of types, spanning from typical land property and savings deposits, to more sophisticated securities, stocks and bonds. The current collateral values supplied, were estimated following an internal proprietary and collateral type-based approach, which is beyond the scope of the analysis in hand. The original dataset was checked for the existence of loans with zero exposure and collaterals that had no or zero value; all such cases of loans and collaterals were excluded. Moreover, any one-to-one cases of connection between loans and collaterals were excluded, since in these

cases all methodologies of Sect. 2 would allocate the same portion of collateral to the unique associated loan; thereby, no added value would be created for the purposes of the methodologies' comparison. The same filtering was performed for the many-to-one and the one-to-many records as well. The final step of the real-world data preparation stage was the isolation of a small total portfolio sub-sample, based on a pseudo-random sampling process. Specifically, the SAS (Statistical Analysis Software) "SURVEYSELECT" procedure [6] was set to provide equal probability simple random accounts sampling, without replacement. Moreover, all accounts and collaterals that did not meet the many-to-many requirement or had not maintained the number of connections had before the aforementioned random sampling were excluded from the final sample. The characteristics of the final sample in terms of total exposures' amount and total collaterals' value (in Euros) along with the minimum and maximum number of connections observed in the loans' and collaterals' sets are provided in Table 1.

From the results provided in Table 2, it is evident that the transportation method allocates the whole available collaterals' value to loans of the real-world portfolio (in a more efficient manner) which has a direct positive impact on the capital requirements.

Table 1 Real-world dataset characteristics

# Loans	# Collaterals	Portfolio total exposure	Total collaterals value
55	55	26,088,448.32	18,381,166.98

# min Loan connections	# max Loan connections	# min Collateral connections	# max Collateral connections
2	9	2	11

Table 2 Results regarding the real-world dataset

Portfolio coverage %			
Collaterals' value/exposures' amount	Proportional allocation	Cutaia method	Transportation method
70.46%	64.57%	64.08%	70.46%

% of loans with coverage >,= or< 100%			
Type of coverage	Proportional allocation	Cutaia method	Transportation method
> 100%	23.64%	21.82%	1.82%
=100%	0.00%	0.00%	63.64%
< 100%	76.36%	78.18%	34.54%

RWA and capital requirements (Euros)			
Amount	Proportional allocation	Cutaia method	Transportation method
RWA	30,710,000	30,774,000	29,942,000
Regulatory Capital	2,456,800	2,461,920	2,395,360

Synthetic Dataset

The dataset consists of a portfolio with 68 loans and a set 65 available collaterals. For the generation of the loan exposures, collateral amounts and loan-collateral connections, the SAS "RAND" function was used (where the input "seed" was set equal to zero, so that the stream of generated numbers is initialized by the computer clock). Moreover, in our implementation each collateral was required to have value no greater than the 50% of the total exposure of its associated loans. A connection between a collateral and a loan was also randomly chosen (based on a Bernoulli trial with a probability $p = 0.05$ of establishing a connection). The characteristics of the dataset so-obtained are provided in Table 3. Finally, Table 4 provides the results regarding the comparison of the three methodologies. From these figures, it is evident that the transportation methodology allocates in a more efficient way the collaterals in the case of synthetic dataset as well.

Table 3 Synthetic dataset: total loans' exposure greater than total collaterals' value

# Loans	# Collaterals	Portfolio total exposure	Total collaterals value
68	65	2,035,600.00	1,828,000.00

# min Loan connections	# max Loan connections	# min Collateral connections	# max Collateral connections
2	5	2	12

Table 4 Results regarding the synthetic dataset

Portfolio coverage %			
Collaterals' value/exposures' amount	Proportional allocation	Cutaia method	Transportation method
89.90%	83.95%	88.48%	89.21%

% of loans with coverage >,= or< 100%			
Type of coverage	Proportional allocation	Cutaia method	Transportation method
> 100%	29.41%	13.24%	0.00%
=100%	4.41%	5.88%	76.47%
< 100%	66.18%	80.88%	23.53%

RWA and capital requirements (Euros)			
Amount	Proportional allocation	Cutaia method	Transportation method
RWA	2,198,900	2,152,800	2,145,400
Regulatory capital	175,912	172,224	171,632

References

1. Basel Committee of Banking Supervision: An Explanatory Note on the Basel II IRB Risk Weight Functions. Bank of International Settlements, Basel (2005)
2. Basel Committee of Banking Supervision: International Convergence of Capital Measurements and Capital Standards: A Revised Framework. Bank of International Settlements, Basel (2006)
3. Cutaia, M.: M2N: optimal collateral to credit allocation. In: Credit Scoring and Credit Control Conference (2011). https://www.business-school.ed.ac.uk/crc/wp-content/uploads/sites/55/2017/03/M2N-Optimal-collateral-to-credit-allocation-Cutaia.pdf
4. Hillier, F.S., Lieberman, G.J.: Introduction to Operations Research, 9th edn. McGraw-Hill International, New Delhi (2010)
5. Hoshen, J., Kopelman, R.: Percolation and cluster distribution. I. Cluster multiple labeling technique and critical concentration algorithm. Phys. Rev. B **14**(8), 3438 (1976)
6. SAS Institute Inc.: SAS/STAT®9.2 User's Guide (2008)

Enhancing Strategic Bidding Optimization for Renewable Energy Auctions: A Risk-Adequate Marginal Cost Model

Chris Stetter, Jan-Hendrik Piel, André Koukal, and Michael H. Breitner

1 Introduction

In recent years, there has been a rapidly increasing number of countries adopting auctions for the allocation of permissions and financial support to renewable energy projects [8]. In Germany, competition for subsidized feed-in compensation for renewable electricity, where project developers participate in auctions and rival for tendered electricity volumes, was introduced with the latest amendment of the Renewable Energy Sources Act (EEG) in 2017. The German auction-mechanism for solar and onshore wind energy is designed as a tender, where project developers compete by specifying their required sales price per unit of electricity (in ct/kWh) as well as a capacity (in MW) to be installed. In the auction process only the most cost-competitive projects with the lowest required financial support are granted until the tendered capacity is reached. From project developers' perspective, competitive pricing in auctions significantly decreases profit margins, which highly increases the sensitivity to risks and uncertainties and thus decreases acceptable valuation errors. In existing literature little support focuses on deriving bid price quantification methods in the realm of renewable energy auctions, in particular from a strategic perspective considering project exogenous factors such as the occurring competition. Some rare examples are the research articles from Anatolitis et al. [1] and Voss and Madlener [4].

An optimal bidding strategy always depends on the country-specific auction design [8]. Such strategies commonly propose to obscure the true cost of a project by adding certain premiums on top of the marginal cost c_t in order to maximize the expected profit. Given the competitive situation and the complex auction design,

C. Stetter (✉) · J.-H. Piel · A. Koukal · M. H. Breitner
Information Systems Institute, Leibniz University Hannover, Hannover, Germany
e-mail: stetter@iwi.uni-hannover.de; piel@iwi.uni-hannover.de; koukal@iwi.uni-hannover.de; breitner@iwi.uni-hannover.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_28

different optimal bidding strategies evolve for the German auction mechanism [1]. It distinguishes between two pricing rules—pay-as-bid and uniform pricing—depending on the source of energy and the participant's legal form. The dominant strategy in a repeated pay-as-bid auction is to choose the bid b that maximizes the expected profit:

$$E(\pi(b)) = \sum_{i=t}^{T} \delta^{i-t} \cdot (b_i - \boldsymbol{c}) \cdot p_i \cdot \prod_{x=1}^{i-t} (1 - p_i) \tag{1}$$

where π is the profit, δ the discount factor and p_i the probability of the bid being successful in round i. In contrast, the weakly dominant strategy for the uniform pricing rule is to bid exactly the agent's marginal cost:

$$b = \boldsymbol{c} \tag{2}$$

Both strategies feature the same starting point of finding an optimal bidding strategy, the marginal cost c, which is the minimum sales price per unit of electricity required to permit an economically viable project construction and operation at an acceptable level of risk. We focus on enhancing the strategic bidding optimization model by proposing a holistic financial modeling approach for a risk-adequate quantification of the marginal cost.

Section 2 presents the proposed methodology. In order to evaluate our model enhancement, we conduct a simulation study of a wind farm in Lower Saxony, Germany in Sect. 3. Finally, conclusions are drawn in Sect. 4.

2 Methodology

Existing strategic bidding optimization models typically consider traditional discounted cash flow models for the quantification of the marginal cost without incorporating project-specific risks and uncertainties and, thus, result in a biased and imprecise bidding strategy. We enhance current estimation approaches for the marginal cost by providing a derivative of the adjusted present value with respect to the sales price per unit of electricity. The presented methodology is based on the model of Piel et al. [3], reformulated as an optimization problem. Its main characteristic is a risk-constrained optimization approach minimizing the required sales price per unit of electricity while considering investment criteria of both equity and debt investors.

Firstly, a probabilistic state-of-the-art cash flow calculation for renewable energy projects is performed. Utilizing Monte Carlo simulations (MCS) allows to simulate uncertain cash flows stemming from identified risks and uncertainties such as capital (CAPEX) or operational expenditures (OPEX). Table 1 presents an income statement and a cash flow statement, which are simulated for each year of the project life cycle $t = (1, \ldots, T_i)$ and each MCS iteration $i = (1, \ldots, I)$. For the sake of simplicity, we only consider the electricity yield $Y_{i,t}$ to be risky in this study. Due to

Table 1 Income and cash flow statements

Income statement		Cash flow statement	
	Revenues		**EBIT**
–	OPEX	–	Taxes on EBIT
=	**EBITDA**	+	Depreciation
–	Depreciation	–	CAPEX
=	**EBIT**	=	**Unlevered free cash flow**

the revenues $R_{i,t} = c \cdot Y_{i,t}$ depending on the electricity yield, it results in probability density function (PDF) estimations for the unlevered free cash flow (FCF), which are the FCF before interest payments are taken into account and serve as the basis for the optimization model.

Secondly, the debt sculpting method is applied to the unlevered FCF in order to optimally utilize the leverage effect of debt financing while a minimum debt service cover ratio (DSCR) is maintained throughout all debt service periods.
The DSCR evaluates the debt service coverage by the cash flow available for debt service and is determined as follows:

$$DSCR_{i,t} = \frac{FCF_{i,t}}{INT_t + P_t}; \quad \forall i \in I, t \in T_{Debt} \tag{3}$$

where INT_t is the interest payment, P_t is the principal repayment, and T_{Debt} is the length of the entire debt service period. Based on a predefined minimum DSCR target β that is maintained throughout all debt service periods at the confidence level $1 - \alpha$, the maximum debt service capacity is derived as follows:

$$DSC_t = \frac{F^{-1}_{FCF,t}(\alpha)}{\beta}; \quad \forall t \in T_{Debt} \tag{4}$$

The interest payments are estimated as the difference of the maximum debt service capacity and the principal repayments, assuming that the debt capital is raised in the form of zero coupon bonds, where r_d is the cost of debt:

$$INT_t = DSC_t - P_t = DSC_t - \frac{DSC_t}{(1 + r_d)^t}; \quad \forall t \in T_{Debt} \tag{5}$$

Thirdly, the adjusted present value (APV) is utilized for evaluating the profitability. Following Myers [2], each iteration of the unlevered FCF is discounted to the valuation date:

$$APV_i = \sum_{t=0}^{T_i} \frac{FCF_{i,t}}{(1 + r_e)^t} + \frac{\tau \cdot INT_t}{(1 + r_d)^t}; \quad \forall i \in I \tag{6}$$

where T_i is the maximum total project life cycle length for all iterations, r_e is the unlevered cost of equity and τ is the corporate tax rate. The use of the APV approach is the best choice, due to its explicit tax-shield consideration [2].

Fourthly, the optimization model minimizes the sales price per unit of generated electricity c under consideration of risk factors, the optimal capital structure and the risk-attitude of involved parties. The latter is determined through the investment criteria of equity investors, by means of the expected APV being zero and debt investors, by means of the DSCR being greater or equal to a target rate β at a certain confidence level $1 - \alpha$. Thus, our optimization problem can be formulated in mathematical terms as follows:

$$\textbf{\textit{Minimize}}\ c \qquad \text{subject to} \tag{7}$$

$$E(f_{APV}) \geq 0 \tag{8}$$

$$F^{-1}_{DSCR,t}(\alpha) \geq \beta; \quad \forall t \in T_{Debt} \tag{9}$$

where f_{APV} is the PDF of the APV and $F^{-1}_{DSCR,t}$ is the inverse cumulative distribution function at percentage point α of the DSCR. As a first step, assuming an initial guess $c_{initial} \in \mathbb{R}^+ \setminus \{0\}$, the cash flow simulation, debt sculpting and present value method are performed. Subsequently, the first derivative of the expected APV with respect to c is determined as follows:

$$\frac{dE(f_{APV})}{dc} = (1-\tau) \cdot \sum_{t=0}^{T} \frac{E(f_{Y,t})}{(1+r_e)^t} + \frac{\tau - \tau^2}{\beta} \cdot \sum_{t=0}^{T_{Debt}} \frac{F^{-1}_{Y,t}(\alpha)}{(1+r_d)^t} \cdot (1-(1+r_d)^{-t}) \tag{10}$$

where $E(f_{Y,t})$ is the expected electricity yield and $F^{-1}_{Y,t}(\alpha)$ the αth percentile of the electricity yield. In a final step, the minimum sales price per unit of generated electricity c that exactly meets the investment criteria of all stakeholders is calculated as follows:

$$c = c_{initial} - \frac{E(f_{APV})}{\frac{dE(f_{APV})}{dc}} \tag{11}$$

3 Case-Study and Results

This section demonstrates the application of our risk-adequate marginal cost model to an onshore wind farm in Lower Saxony, Germany using a prototypical implementation in Python. Table 2 presents its project characteristics [5–7].

The electricity yield is calculated as follows using the annual Weibull wind speed PDF $f_{Weibull,i,t}(v)$, which was estimated using NASA's MERRA-2 dataset:

$$Y_{i,t} = \int_{v=0}^{V} f_{Weibull,i,t}(v) \cdot P(v) dv \cdot NOH \cdot \delta; \quad \forall i \in I, t \in T_i \tag{12}$$

Table 2 Project characteristics of the onshore wind farm project under investigation

Wind turbines	5 Nordex N131-3.3	Project duration	20 years
Total capacity	16.5 MW	CAPEX	1513 €/kW
Farm efficiency	87%	OPEX (Year 1–10)	21.71 €/MWh
Net operating hours	7.635 h/turb.	OPEX (Year 11–20)	23.60 €/MWh
Corporate tax	30%	Unlev. cost of equity	3.18%
Straight line depreciation	16 years	Cost of debt	3.5%
DSCR target	1.2 at $1-\alpha$=75%	Debt service period	14 years

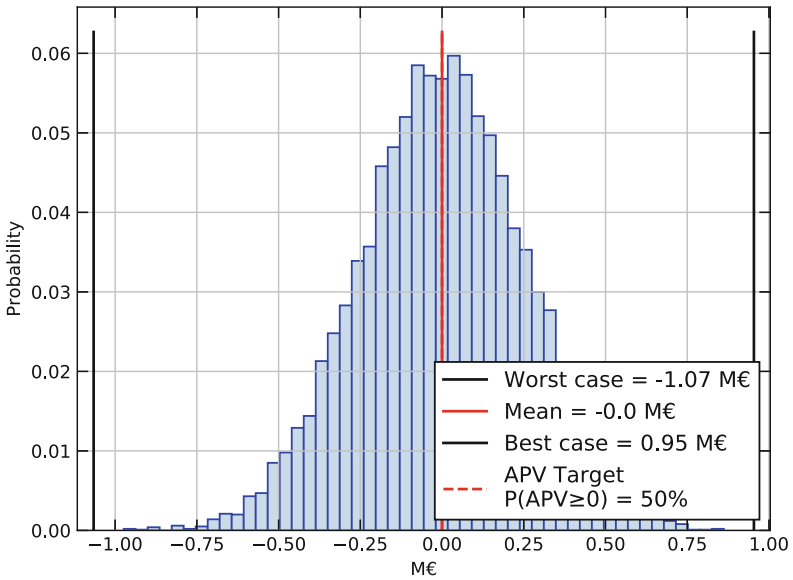


Fig. 1 Histogram of adjusted present value after optimization

where $P(v)$ is the turbine's cumulative power curve, V is the cut-out wind speed, NOH is the net operating hours and δ is the farm efficiency.

Applying our model with 10,000 MCS iterations to the fictive onshore wind farm yields marginal cost of 5.97 ct/kWh. Figure 1 demonstrates that the decision criteria of equity capital investors is fulfilled on average as the expected value of the APV is equal to or greater than zero. Likewise, a project is considered financially viable from lenders' perspective if a target ratio is covered for in every debt service period with a certain probability. Figure 2 shows that for every debt service period, the mean DSCR is equal to or greater than the target of 1.2 in 75% of the iterations. Consequently, the investment criteria of both equity and debt investors are exactly fulfilled at the estimated marginal cost.

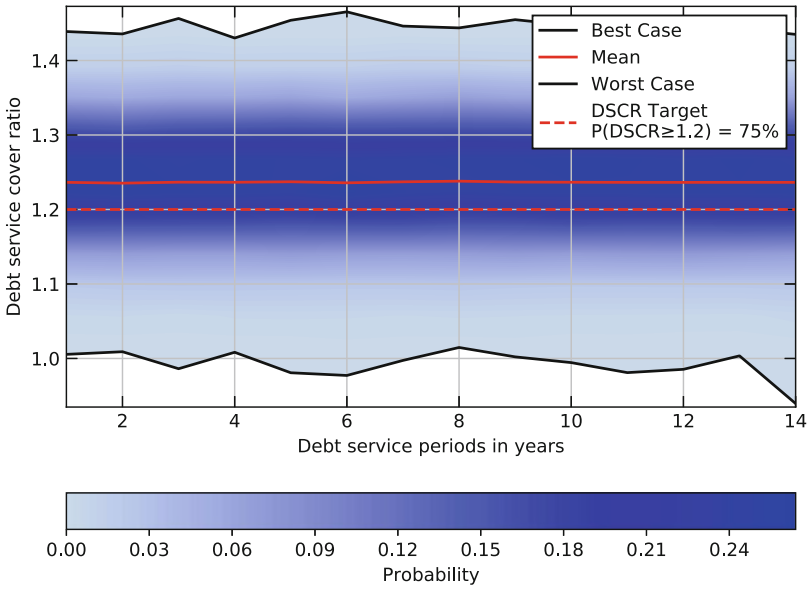


Fig. 2 Debt service coverage after optimization

4 Conclusion

Determining bid prices that are cost-competitive and sustainable in terms of a likely project realization is a major challenge that results from the shift toward auction mechanisms. It follows, that the investment criteria of all project stakeholders comprising equity and debt investors should be considered in the bid price quantification process. We have proposed an enhanced, risk-adequate estimation approach for the marginal cost, which is the starting point for strategic bidding optimization, by providing a derivative of the adjusted present value with respect to the sales price per unit of electricity. For the analyzed project it was demonstrated that the quantified marginal cost are sustainable due to the investment criteria being fulfilled, and at the same time cost-competitive as it is below the average market clearing price of 6.29 ct/kWh of the three last German onshore wind auctions. Hence, starting from the marginal cost, the simulated project developer would have had additional room for bidding strategically.

In contrast to the levelized cost of electricity, which has a similar meaning [3], our approach not only allows for the consideration of risk factors as well as specific project finance characteristics. Most importantly, as our modelling approach permits the direct quantitative incorporation of investment criteria within strategic auction bids, we contribute to an enhanced strategic bidding optimization and comprehensive methodological support for project developers in competitive renewable energy auctions. Our introduced modeling approach aims at accounting

for the risk-attitude of all involved parties, which is indispensable for successful realizations of future renewable energy projects.

References

1. Anatolitis, V., Welisch, M.: Putting renewable energy auctions into action–an agent-based model of onshore wind power auctions in Germany. Energy Policy **110**, 394–402 (2017)
2. Myers, S.C.: Interactions of corporate financing and investment decisions–implications for capital budgeting. J. Financ. **29**(1), 1–25 (1974)
3. Piel, J.H., Hamann, J.F., Koukal, A., Breitner, M.H.: Promoting the system integration of renewable energies: toward a decision support system for incentivizing spatially diversified deployment. J. Manag. Inf. Syst. **34**(4), 994–1022 (2017)
4. Voss, A., Madlener, R.: Auction schemes, bidding strategies and the cost-optimal level of promoting renewable electricity in Germany. In: FCN Working Paper (2017)
5. Wallasch, A.K., Lüers, S., Rehfeldt, K.: In: Weiterbetrieb von Windenergieanlagen nach 2020, Deutsche WindGuard, Tech. Rep (2016)
6. Wallasch, A.K., Lüers, S., Rehfeldt, K.: In: Kostendruck und Technologieentwicklung im Zuge der ersten Ausschreibungsrunden für die Windenergie an Land, Deutsche WindGuard, Tech. Rep (2017)
7. Wallasch, A.K., Lüers, S., Rehfeldt, K.: Wirtschaftlichkeit unterschiedlicher Nabenhöhen von Windenergieanlagen, Deutsche WindGuard, Tech. Rep (2017)
8. Winkler, J., Magosch, M., Ragwitz, M.: Effectiveness and efficiency of auctions for supporting renewable electricity–what can we learn from recent experiences? Renew. Energy **119**, 473–489 (2018)

An Optimization Model for Multi-Asset Batch Auctions with Uniform Clearing Prices

Tom Walther

1 Introduction

In continuous-time asset exchange mechanisms, orders are typically collected in order books of two assets that are traded against each other. A trade happens whenever at least two matching orders are available, i.e., if there exists an exchange rate satisfying their specified limit prices. Most real-world asset exchanges rely on some principal asset (e.g., USD) and only offer trading pairs with this asset. Hence, arbitrary assets can not be traded directly against each other, thus limiting trading opportunities and incurring unnecessary costs for market participants. This potentially affects any market that trades financial assets, as for instance foreign exchange markets.

In our approach, we want to collect orders for a set of multiple assets in a single joint order book and compute exchange prices for all asset pairs simultaneously at the end of discrete time intervals (*multi-asset batch auction*). Trades between the same asset pairs are then all executed at the same exchange rate (*uniform clearing price*). This mechanism enables so-called *ring trades*, where orders are matched along cycles of assets, thus enhancing liquidity for traders. In order to exclude arbitrage opportunities, we require prices to be consistent along such cycles, i.e., we want to impose the constraint

$$p_{i|j} \cdot p_{j|k} = p_{i|k} \tag{1}$$

for the prices of all assets i, j, k.

T. Walther (✉)
Gnosis/Zuse Institute Berlin, Berlin, Germany
e-mail: tom.walther@gnosis.pm

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_29

Our main motivation for considering this type of asset exchange mechanism stems from the emerging blockchain technology, where no intermediary is required in order to freely exchange blockchain-based assets (so-called *tokens*) among each other. Current centralized blockchain asset exchanges do not make full use of this potential and merely try to replicate conventional stock exchanges, thus incurring the same frictions and arbitrage opportunities mentioned above.

A good overview on the theory of arbitrage from a finance perspective is given in [4]. The concept of frequent batch auctions aiming at reducing arbitrage and improving liquidity has been investigated extensively in [2]. Moreover, the advantages of uniform-price clearing have been discussed in [3]. In this document, we want to describe the problem of determining uniform clearing prices for all pairs of assets involved in a multi-asset batch auction process, and present a mixed-integer programming (MIP) solution approach.

2 Problem Statement

Data Let $\mathcal{A} := \{\alpha_0 \dots \alpha_{n-1}\}$ denote the set of the n assets that are being considered. Without loss of generality, the asset α_0 shall be referred to as *reference asset*. Moreover, let there be a set $\mathcal{O} = \{\omega_1 \dots \omega_N\}$ of N limit buy orders in the batch to be processed. Every such buy order consists of a tuple $(i, j, \overline{x}, \pi)$ that is to be read as

> *"Buy (at most)* $\overline{x}$ *units of asset* α_i *for asset* α_j *if the rate* $p_{i|j}$ *is at most* π*"*.

Please note that limit sell orders can be modelled similarly, but will not be considered here in order to keep notation as simple as possible.

Variables The pairwise exchange rates between two assets α_i and α_j are denoted by $p_{i|j}$, meaning the price of one unit of α_i measured in units of α_j. As an example, if $p_{i|j} = 10$, one would need to pay an amount of 10 units of α_j in order to purchase a unit of α_i. In the optimization, we want to determine all pairwise exchange rates between assets as well as the fraction of every order that can be fulfilled at these rates. Obviously, this fraction may only be positive for an order if the respective exchange rate satisfies the given limit price.

Objective Our goal is to find exchange rates for all asset pairs that maximize the trade volume that is enabled, measured in units of the reference asset.

Constraints The solution we are aiming at needs to satisfy several requirements:

(a) *buy limit price*—for every buy order $\omega = (i, j, \overline{x}, \pi) \in \mathcal{O}$: The order can only be executed (fully or fractionally) if the exchange rate does not exceed the stated limit price, i.e., $p_{i|j} \leq \pi$.
(b) *asset balance*—for every asset $\alpha_i \in \mathcal{A}$: The amount of assets α_i bought must equal the amount of assets α_i sold across all orders.

(c) *price coherence*—for all asset pairs (α_i, α_j): $p_{i|j} \cdot p_{j|i} = 1$.
(d) *arbitrage-freeness*—for all asset triples $(\alpha_i, \alpha_j, \alpha_l)$: $p_{i|j} \cdot p_{j|l} = p_{i|l}$.

3 Mixed-Integer Programming Model

3.1 Model Data

With the exchange rates between pairs of assets being variables, modelling arbitrage-freeness intuitively leads to a nonlinear formulation due to the multiplications that are required. This would result in tight limitations to the tractable problem size as well as numerical instability. However, it is possible to avoid nonlinearity by not considering all pairwise asset rates explicitely but representing all asset prices only w.r.t. the reference asset α_0. Let $p_i := p_{i|0}$ denote the price of asset α_i expressed in units of α_0 (hence, $p_0 = 1$). Applying the arbitrage-freeness and price coherence conditions directly, we can write the exchange rate between two assets α_i and α_j as

$$p_{i|j} = p_{i|0} \cdot p_{0|j} = \frac{p_{i|0}}{p_{j|0}} = \frac{p_i}{p_j}. \tag{2}$$

In order to be able to state a MIP formulation for our problem, we will express all data and variables in terms of matrices and vectors. First, we store all asset price variables in a vector $(p_i) =: \mathbf{p} \in \mathbb{R}^n_{\geq 0}$. It makes sense to set an explicit lower and upper bound for the price of every asset α_i (e.g., in order to avoid excessive fluctuations), so let us require $p_i \in [\underline{p}_i, \overline{p}_i]$. These bounds could for example be derived as half/double the previous asset prices, or simply be set to some low/high values that appear reasonable. In order to ensure $p_0 = 1$ for the reference asset α_0, we set $\underline{p}_0 = \overline{p}_0 = 1$. The price bounds shall be stored in vectors $\underline{\mathbf{p}}$ and $\overline{\mathbf{p}}$, respectively.

Moreover, we introduce two data matrices

$$\mathbf{T}^b \in \{0,1\}^{N \times n} \quad \text{with} \quad \mathbf{T}^b \ni t^b_{k,i} = 1 \Leftrightarrow \text{asset } \alpha_i \text{ to be bought in order } \omega_k,$$

$$\mathbf{T}^s \in \{0,1\}^{N \times n} \quad \text{with} \quad \mathbf{T}^s \ni t^s_{k,i} = 1 \Leftrightarrow \text{asset } \alpha_i \text{ to be sold in order } \omega_k.$$

Since we are only considering orders of one asset against one other, there must be exactly one entry equal to 1 per row (order) in both $\mathbf{T}^b$ and $\mathbf{T}^s$. The maximum number of units of the asset α_i to be bought in an order $\omega_k = (i, j, \overline{x}, \pi)_k$ shall be denoted by $\overline{x}_k$ and stored, for all orders, in a vector $\overline{\mathbf{x}} \in \mathbb{R}^N_{\geq 0}$. The precise buy amount of α_i, which is to be determined by the optimization procedure, will be denoted by $(x_k) =: \mathbf{x} \in \mathbb{R}^N_{\geq 0}$. Hence, $x_k \leq \overline{x}_k$ for all orders $\omega_k \in \mathcal{O}$. However, we will only determine the values in $\mathbf{x}$ indirectly. Instead, for the purpose of keeping the model linear, we define another variable vector $(v_k) =: \mathbf{v} \in \mathbb{R}^N_{\geq 0}$, where

v_k represents the traded volume of assets in the order ω_k denoted in units of the reference asset α_0. The limit prices of all orders are stored as vector $(\pi_k) =: \boldsymbol{\pi} \in \mathbb{R}^N_{\geq 0}$, where $\pi_k \in \boldsymbol{\pi}$ refers to the exchange rate between the respective assets of order ω_k according to the definition above.

Besides the main problem variables $\mathbf{p} \in \mathbb{R}^n_{\geq 0}$ and $\mathbf{v} \in \mathbb{R}^N_{\geq 0}$, our MIP model requires several additional variables. Most importantly, let $\mathbf{z} \in \{0, 1\}^N$ be a vector of binary variables, where $z_k \in \mathbf{z}$ indicates whether an order ω_k may be (fully or partially) executed, or not. Precisely, we require $z_k = 1$ if and only if the exchange rate between the assets of ω_k satisfies the respective limit price π_k, otherwise $z_k = 0$. Furthermore, the feasible region for the execution volume v_k of an order ω_k depends on the value of z_k. In particular, v_k must be set to zero if $z_k = 0$ and can only be non-zero otherwise. In order to model this disjoint behaviour, we make use of a *disjunctive programming* formulation [1] for which we introduce the auxiliary price variable vectors $\mathbf{p}^{b,0}, \mathbf{p}^{b,1}, \mathbf{p}^{s,0}, \mathbf{p}^{s,1} \in \mathbb{R}^N_{\geq 0}$. The idea behind this approach will be expanded upon after the model formulation.

Our model allows for setting a minimum fraction of execution for every order, when its limit price is satisfied, as a parameter. In this paper, we will use a global value $\underline{r} \in [0, 1]$ for all orders.

3.2 Model Formulation

With all these definitions, we can finally state the MIP model for the problem of multi-asset batch auctions with uniform clearing prices as follows:

$$\max_{\mathbf{v},\mathbf{p}} \quad \sum_{k=1}^{N} v_k \tag{3a}$$

$$\text{s.t.} \quad \sum_{k=1}^{N} t^b_{k,i}\, v_k = \sum_{k=1}^{N} t^s_{k,i}\, v_k \qquad \forall\, i = 0 \ldots n-1 \tag{3b}$$

$$\sum_{i=1}^{n-1} t^b_{k,i}\, \underline{p}_i (1 - z_k) \leq p^{b,0}_k \leq \sum_{i=1}^{n-1} t^b_{k,i}\, \overline{p}_i (1 - z_k) \quad \forall\, k = 1 \ldots N \tag{3c}$$

$$\sum_{i=1}^{n-1} t^s_{k,i}\, \underline{p}_i (1 - z_k) \leq p^{s,0}_k \leq \sum_{i=1}^{n-1} t^s_{k,i}\, \overline{p}_i (1 - z_k) \quad \forall\, k = 1 \ldots N \tag{3d}$$

$$\sum_{i=1}^{n-1} t^b_{k,i}\, \underline{p}_i z_k \leq p^{b,1}_k \leq \sum_{i=1}^{n-1} t^b_{k,i}\, \overline{p}_i z_k \qquad \forall\, k = 1 \ldots N \tag{3e}$$

$$\sum_{i=1}^{n-1} t_{k,i}^{s}\, \underline{p}_i z_k \leq p_k^{s,1} \leq \sum_{i=1}^{n-1} t_{k,i}^{s}\, \overline{p}_i z_k \qquad \forall\, k = 1 \ldots N \tag{3f}$$

$$\sum_{i=1}^{n-1} t_{k,i}^{b}\, p_i = p_k^{b,0} + p_k^{b,1} \qquad \forall\, k = 1 \ldots N \tag{3g}$$

$$\sum_{i=1}^{n-1} t_{k,i}^{s}\, p_i = p_k^{s,0} + p_k^{s,1} \qquad \forall\, k = 1 \ldots N \tag{3h}$$

$$p_k^{b,0} \geq \pi_k\, p_k^{s,0} \qquad \forall\, k = 1 \ldots N \tag{3i}$$

$$p_k^{b,1} \leq \pi_k\, p_k^{s,1} \qquad \forall\, k = 1 \ldots N \tag{3j}$$

$$v_k \geq \underline{r}\, \overline{x}_k\, p_k^{b,1} \qquad \forall\, k = 1 \ldots N \tag{3k}$$

$$v_k \leq \overline{x}_k\, p_k^{b,1} \qquad \forall\, k = 1 \ldots N \tag{3l}$$

$$\mathbf{z} \in \{0, 1\}^N$$

$$\mathbf{v},\ \mathbf{p}^{b,0},\ \mathbf{p}^{b,1},\ \mathbf{p}^{s,0},\ \mathbf{p}^{s,1} \in \mathbb{R}^N_{\geq 0}$$

$$\mathbf{p} \in \mathbb{R}^n_{\geq 0}$$

The objective function (3a) maximizes the total trading volume in terms of units of the reference asset α_0 that is processed with all orders. Constraint (3b) secures that the total buy and sell volumes across all orders must be equal for every asset. With uniform clearing prices, volume balance implies asset balance. Please note that the summation in this constraint (as well as in the following ones) is needed to select the correct volume variables for the respective asset under consideration. The auxiliary variables in $\mathbf{p}^{b,0}$ and $\mathbf{p}^{s,0}$ refer to the prices of the buy- and sell-asset of every order ω_k if that order is not to be executed ($z_k = 0$). In that case, their values must lie within the respective bounds provided by $\underline{\mathbf{p}}$ and $\overline{\mathbf{p}}$, and otherwise shall be set to zero. This requirement is ensured by the constraints (3c) and (3d). Similarly as above, the constraints (3e) and (3f) control the auxiliary variables in $\mathbf{p}^{b,1}$ and $\mathbf{p}^{s,1}$ for the case that an order ω_k is executable ($z_k = 1$). The relation between the auxiliary price variables and the actual asset prices is established by the constraints (3g) and (3h). For all orders, the disjunctive behaviour is modelled by the constraints (3i)–(3l). The idea is as follows: If some order ω_k is not to be executed ($z_k = 0$), then the prices $p_k^{b,0}$ and $p_k^{s,0}$ must both lie within the bounds given by (3c) and (3d) as well as fulfill (3i) (i.e., not satisfy the limit price). At the same time, $p_k^{b,1}$

and $p_k^{s,1}$ are set to zero by (3e) and (3f), thus trivially satisfying (3j). This then also implies the volume v_k to be set to zero by (3k) and (3l). Conversely, if ω_k is to be executed ($z_k = 1$), $p_k^{b,0}$ and $p_k^{s,0}$ are set to zero by (3c) and (3d), while $p_k^{b,1}$ and $p_k^{s,1}$ lie within the bounds provided by (3e) and (3f) as well as fulfill (3j) (i.e., satisfy the limit price). This finally requires the execution volume v_k to be within the specified fractions via the constraints (3l) and (3l).

3.3 Computational Results and Outlook

In order to investigate the performance of our model, we have conducted computational experiments for different numbers of assets ($n \in \{5, 10, 20, 50\}$) and orders ($N \in \{100, 200, 500\}$). For every combination of n and N, we have generated 20 random instances, whereby the randomness reflects our expectation of somewhat realistic situations. In particular, we expect not all assets to be equally important in terms of trading volume and, hence, to have varying numbers of orders on different asset pairs. We used Gurobi 8.0.0 as MIP solver on an Intel(R) Core(TM) i7-8550U CPU @ 1.80 GHz machine with 16 Gb RAM and using four threads, and a timelimit set to 1800 s for every instance.

The results of our computational experiments as in Table 1 show that runtimes of the MIP formulation sharply increase both with the number of assets and the number of orders that are being considered. Several instances with more than 20 assets and more than 200 orders could not be solved to optimality within the timelimit. The largest instances with 50 assets and 500 orders even left very substantial gaps in most cases. If such larger problems are to be considered, we can think of the following heuristics/approximations:

Table 1 Computational results of instances of the MIP model (3)

# orders		# assets			
		5	10	20	50
100	∅ runtime	0.32	0.32	0.90	5.54
	# timeouts	0	0	0	0
	– ∅ gap	–	–	–	–
200	∅ runtime	1.31	2.49	26.25	571.10
	# timeouts	0	0	0	12
	– ∅ gap	–	–	–	23.46%
500	∅ runtime	15.67	82.47	522.72	–
	# timeouts	0	0	11	20
	– ∅ gap	–	–	21.59%	74.82%

The (geometric) means of the runtimes have been computed only with respect to the instances that could be solved to optimality before the timelimit. Conversely, the average optimality gap does not take solved instances into account

– Aggregate orders with similar limit prices on every asset pair
– Optimize over subsets of assets separately and fix prices in overall problem

Finally, a possible direction for future research besides finding more efficient solution methods for the problem at hand would be the consideration of more complex order types. For example, instead of trading one single asset against one other, traders might want to trade a set of multiple assets against multiple others (*basket orders*), without being interested in individual exchange rates. Another possibility could be to allow the order amounts to be flexible with the determined prices, i.e., traders might want to buy/sell more of some asset if its price is favorable.

References

1. Balas, E.: Disjunctive programming. Ann. Discrete Math. **5**, 3–51 (1979). https://doi.org/10.1016/S0167-5060(08)70342-X
2. Budish, E., Cramton, P., Shim, J.: The high-frequency trading arms race: frequent batch auctions as a market design response. Q. J. Econ. **130**(4), 1547–1621 (2015). https://doi.org/10.2139/ssrn.2388265
3. Engelbrecht-Wiggans, R., Kahn, C.: Multi-unit auctions with uniform prices. Econ. Theory **12**, 227–258 (1998). https://doi.org/10.1007/s001990050220
4. Ross, S.A.: Neoclassical Finance, Princeton Lectures in Finance, Chapter 1. Princeton University, Princeton (2009)

Part VII
Graphs and Networks

A Network Flow Formulation of the Location-Scheduling Problem for Electric Vehicle Charge Stations with Incomplete Information

Peter Czimmermann and Ľuboš Buzna

1 Introduction

Over the last years, optimization has often been proposed to support the design of charging infrastructure. The special class of models was developed to cover trajectories of vehicles [2, 6]. This approach is applicable in the design of the charging infrastructure along motorways to cover long distance trips. Customized approaches have also been proposed for urban areas. In [1] an approach was proposed to deal with covering hexagon shaped geographical areas.

Here, we build on our previous work [4, 5], where we proposed a location-scheduling formulation of the problem to determine suitable positions of charge stations, while assuming that the operation of the fleet remains unaffected, i.e. for the recharging of batteries only parking events are used. We propose an extension of this approach by suggesting a network flow formulation of the problem and by discussing the role of available information when electric vehicle drivers choose a charging point.

2 Problem Formulation

The fleet of vehicles is represented by the set C and each vehicle is equipped with a battery of capacity β. The set of candidate locations to locate charge stations is denoted by I. We discretise the time by dividing it into the set T of non-overlapping intervals of equal size. For each vehicle $c \in C$, we extract from the GPS data an

P. Czimmermann (✉) · Ľ. Buzna
University of Žilina, Žilina, Slovakia
e-mail: Peter.Czimmermann@fri.uniza.sk

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_30

ordered sequence of parking events R_c, and N_{cr} denotes the list of time intervals $t \in T$ that have an overlap with the parking event $r \in R_c$. To simplify notation, for each vehicle, we add to R_c the fictional introductory parking event 0 and fictional terminal parking event r_c. From the GPS data, we extracted driving distances and we estimated the energy consumption when vehicle $c \in C$ drives towards stop $r \in R_c$ as $w_{rc} \geq 0$. Constant $a_{ct} \in \langle 0, 1\rangle$ is the fraction of the time interval $t \in T$ when vehicle $c \in C$ is parking. The charging speed of all charge points is denoted by ρ. (We suppose in this version of the model that there is an identical charging speed at all charging points.) To simplify the description of the model, we define $B_{itc} \in \{0, 1\}$, where $B_{itc} = 1$ if vehicle $c \in C$ is parking at location $i \in I$ to be charged during the time interval $t \in T$, and $B_{itc} = 0$ otherwise.

Decisions are described by the following variables:

- $s_i \in Z^+$ represents the number of charge points allocated to station $i \in I$,
- $x_{ct} \in \{0, 1\}$, if vehicle $c \in C$ is being charged during the time interval $t \in T$, then $x_{ct} = 1$ and $x_{ct} = 0$ otherwise,
- $d_{cr} \geq 0$ corresponds to the state of charge of the vehicle $c \in C$ at the beginning of the parking event $r \in R_c$.

We consider the objective function to minimize the number of charge points to be established, i.e.:

$$\text{Minimize} \sum_{i \in I} s_i \tag{1}$$

$$\text{subject to} \sum_{c \in C} B_{itc} x_{ct} \leq s_i \qquad \text{for } i \in I, t \in T \tag{2}$$

$$d_{c0} \leq \alpha\beta \tag{3}$$

$$d_{cr} + \sum_{t \in N_{c,r}} a_{ct} x_{ct} \rho \leq \beta \qquad \text{for } c \in C, r \in R_c \cup \{r_c\} \tag{4}$$

$$d_{cr} \leq d_{c,r-1} - w_{rc} + \sum_{t \in N_{c,r-1}} a_{ct} x_{ct} \rho \qquad \text{for } c \in C, r \in R_c \cup \{r_c\} \tag{5}$$

The set of constraints (2) limits the number of used charge points. The initial upper limit for the driving distance is set in constraints (3) to $\alpha\beta$, where $\alpha \in \langle 0, 1\rangle$ is an appropriate constant. Constraints (4) ensure that battery capacity is never exceeded and constraints (5) maintain continuity in the charging and discharging of batteries.

2.1 Results of Numerical Experiments

Optimization problem (1)–(5) assumes that information about the future trips and future charging events of all vehicles is available when the decisions about the

Table 1 The number of vehicles in use during the solving of the optimization problem (1)–(5) and the number of feasible vehicles obtained by applying evaluation procedures described in [4]

	Number of vehicles		
	Input of the	Output of the procedure of	
Benchmarks	optimization problem	coordinated charging	uncoordinated charging
Benchmark 1	609	401	363
Benchmark 2	1186	652	536
Benchmark 3	1287	689	586
Benchmark 4	1102	592	589

charging of electric vehicles are taken. This is an unrealistic assumption and it may lead to an underestimation of the number of charging points that can recharge all vehicles in the fleet. In [4], we have proposed two strategies to evaluate the number of vehicles that can be served by a charging infrastructure: coordinated charging and uncoordinated charging. The strategy of coordinated charging assumes that drivers posses information about charge points and know when the charge stations that are occupied will be freed, thus, drivers can select those charge points where the batteries can be charged to the maximum capacity. Moreover, drivers are assumed to be cooperative, i.e., they unplug the vehicle when the charging is terminated. When applying the strategy of uncoordinated charging, drivers choose the charge station $i \in I$ with the largest number of free charge points. The battery level of each vehicle is updated and the charge point is freed when the vehicle departs for the next trip.

Using real-world data describing the operation of a large fleet of taxicabs collected within the city of Stockholm, we have done extensive numerical experiments [5]. In these experiments we compared the number of vehicles (elements of C) for which the charging infrastructure is the optimal solution of the problem (1)–(5), and in accordance with the above mentioned evaluation procedures [4]. Typical results are shown in Table 1. In Column 2, there is the number of vehicles in the fleet. These vehicles belong to the input of the optimization problem. The output of (1)–(5) is the charging infrastructure (located charging points). If we apply this charging infrastructure to the evaluation procedures, then we see that not every vehicle in the fleet can be recharged (Columns 3, 4 in Table 1). We observe that the optimization problem systematically underestimates the number of charging points that are necessary to serve all the fleet of electric vehicles.

3 Network Flow Problem

In [3], we introduced a flow model approximating the problem of minimize the number of charge stations. In this section we amend the flow model to be approximation of the model (1)–(5). More precisely, we consider its discrete version, where d_{cr} are described by the set of discrete states and $a_{ct} \in \{0, 1\}$.

We need to define some subsets of $T = \{0, 1, \ldots, |T|\}$. Let $c \in C$ be a vehicle, $i \in I$ be a candidate location, and $j \in \{1, \ldots, M\}$ possible charging point in i ($M = \max\{\sum_{c \in C} B_{itc} : i \in I, t \in T\}$). The set $T_c = N_c \cup \{\max(N_{cr}) + 1 | r = 0, 1, \ldots, r_c - 1\}$ (where $N_c = N_{c0} \cup N_{c1} \cup \cdots \cup N_{cr_c}$), and $T'_{ij} \subset T$, where $t \in T'_{ij}$ if and only if $\sum_{c \in C} B_{itc} > 0$ or $t = 0$. We also suppose that $a_{ct} \in \{0, 1\}$. We construct a network G. The set of vertices of G is the union of the sets:
$V_1 = \{v_{ct} : c \in C, t \in T_c\} \cup \{v'_{ct} : c \in C, t \in N_c\}$, if $a_{ct} = 1$, then v_{ct} and v'_{ct} represent the parked vehicle c in time t, otherwise $a_{ct} = 0$, then v_{ct} represents all time intervals between parking events r and $r + 1$ (when the vehicle is not available for charging), where $t = \max(N_{cr}) + 1$,
$V_2 = \{u_{ijt}, u'_{ijt} : i \in I, j \leq M, t \in T'_{ij}\}$, where u_{ijt}, u'_{ijt} represent the candidate for charge point j in location i and time t (u_{ijt}, u'_{ijt} are defined only if there is a parked vehicle in $i \in I$, i.e. $\sum_{c \in C} B_{itc} > 0$),
$V_3 = \{v_s, v_z\}$, where v_s is the source and v_z is the sink.

The set of edges of G is the union of the sets:
$E_1 = \{(v_s, v_{c0}) : c \in C\}$,
$E_2 = \{(v_s, u_{ij0}) : i \in I, j \leq M\}$,
$E_3 = \{(v_{ct}, v_{c\tau}), (v_{cn}, v_z) : c \in C, \tau = \min\{\bar{t} \mid \bar{t} > t \wedge \bar{t} \in T_c\}, n = \max(T_c)\}$,
$E_4 = \{(u_{ijt}, u_{ij\tau}), (u_{ijn}, v_z) : i \in I, j \leq M, \tau = \min\{\bar{t} \mid \bar{t} > t \wedge \bar{t} \in T'_{ij}\}, n = \max(T'_{ij})\}$
$E_5 = \{(u_{ijt}, u'_{ijt}) : i \in I, j \leq M, t \in T'_{ij}\}$,
$E_6 = \{(u'_{ijt}, v'_{ct}) : i \in I, j \leq M, c \in C, t \in N_c \cap T'_{ij}, B_{itc} = 1\}$,
$E_7 = \{(v'_{ct}, v_{ct}) : c \in C, t \in N_c\}$,
$E_8 = \{(v_{ct}, v_z) : c \in C, r \in R_c - \{r_c\}, t = \max(N_{cr}) + 1\}$.

Every edge e has assigned to it a lower and upper bound for the flow, which are represented by the interval $\langle l(e), u(e)\rangle$. These bounds are given in Table 2 (Fig. 1).

The network is defined in such a way that the problem of minimizing the number of charge points (in its discrete version) is equivalent to the problem of finding feasible flow with a minimum number of non-zero flows on edges (v_s, u_{ij0}). This can be called the sparsest feasible flow problem.

Table 2 Lower and upper bounds for the flows on edges

E_1	$\langle 0, \alpha\beta\rangle$	E_5	$\langle 0, \rho\rangle$
E_2	$\langle 0, K_\infty\rangle$, where K_∞ is a large constant	E_6	$\langle 0, \rho\rangle$
E_3	$\langle 0, \beta\rangle$	E_7	$\langle 0, \rho\rangle$
E_4	$\langle 0, K_\infty\rangle$, where K_∞ is a large constant	E_8	$\langle w_{rc}, w_{rc}\rangle$, where $c \in C, r \in R_c$

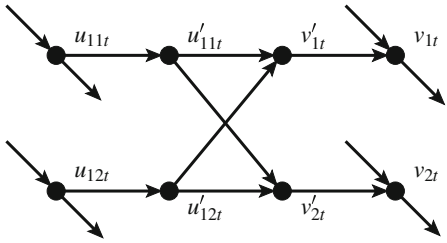


Fig. 1 Part of the network for $t \in T_1 \cap T_2 \cap T'_{11} \cap T'_{12}$

4 Solving Method

In the previous section, we stated the problem of finding the feasible flow with a minimum number of nonzero flows on given edges. The sparsest feasible flow can be found in small cases by the modified branch and bound method. We can also use an iterative heuristic from [3] that uses augmenting paths in networks.

The Augmenting Path in network G with flow f is a (half-)path P such that the reserve $r(e) > 0$ for any edge $e \in P$. The reserve of the edge $e \in P$ is $r(e) = u(e) - f(e)$ when e is a forward edge in P, and $r(e) = f(e) - l(e)$ when e is a reverse edge in P. The reserve of the augmenting path is $r = \min\{r(e)|e \in P\}$.

However, we need to consider the fact that the operator (and drivers) must work with incomplete information. Hence we suggest the following approach.

1. We find a minimum feasible flow f in G.
2. For $t = 1, 2, \ldots, |T|$, we do: $\forall u_{ijt}, u_{klt}$, we find an augmenting path of the form F_1 or F_2 (described below).

2a. If there is an augmenting path P of the form F_1 from u_{ijt} to u_{klt} and reserve $r > 0$, then we change the flow in G in the following way: if $e \in P$ is the forward edge in P, then the flow is increased by the value r; if $e \in P$ is the reverse edge in P, then the flow is decreased by the value r; every edge from the path $v_s, u_{ij0}, \ldots, u_{ijt}$ has increased the flow by r; every edge from the path $v_s, u_{kl0}, \ldots, u_{klt}$ has decreased the flow by r.

2b. If there is an augmenting path P of the form F_2 from u_{ijt} to u_{ilt} (and there is no path of the form F_1 from u_{ijt}) and reserve $r > 0$, then we change the flow in G in the following way: if $e \in P$ is the forward edge in P, then the flow is increased by the value r; if $e \in P$ is the reverse edge in P, then the flow is decreased by the value r; every edge from the path $v_s, u_{ij0}, \ldots, u_{ijt}$ has increased the flow by r; every edge from the path $v_s, u_{il0}, \ldots, u_{ilt}$ has decreased the flow by r.

Augmenting Path of the Form F_1 represents the situations, in which driver can make decision about the charging only one step (one drive) before. Let the flow entering the vertex u_{ijt} be greater than the flow entering the vertex u_{klt}. Let there be a vehicle c such that $t = \max N_{rc}$. We find an augmenting path of

the form

$$u_{ijt} \to u'_{ijt} \to v'_{ct} \to v_{ct} \to v_{ct_1} \to v_{ct_2} \leftarrow v'_{ct_2} \leftarrow u'_{klt_2} \leftarrow u_{klt_2} \leftarrow \ldots$$
$$\leftarrow u_{klt'} \leftarrow \cdots \leftarrow u_{klt},$$

where $t_1, t_2 \in T_c$, $t_1 = t + 1$, $t_2 = \min N_{c,r+1}$, and $t' = t'_1, \ldots, t'_x$, where $t < t'_1 < \cdots < t'_x < t_2$ are consecutive time intervals from the set T'_{kl}.

Augmenting Path of the Form F_2 helps us to reduce the number of charge points in each location. Let $i = k$ and $j < l$. Let there be a vehicle c such that $t \in N_{cr}$. We find an augmenting path P of the form $u_{ijt} \to u'_{ijt} \to v'_{ct} \leftarrow u'_{ilt} \leftarrow u_{ilt}$.

5 Conclusions

In our contribution, we showed that the location-scheduling problem of finding the appropriate design of charging infrastructure can be approximated by the problem of finding the sparsest feasible flow in the network. The advantage of the flow formulation is that it allows us to model various levels of information about charging possibilities available to drivers and operators. We would like to use the flow representation of the problem to analyze the relationship between the effectiveness of the system and the minimum amount of information known by operators and drivers. This approach provides several new possibilities, which we would like to study in the future.

Acknowledgements This work was supported by the research grants VEGA 1/0089/19, APVV-15-0179 and it was facilitated by the FP 7 project ERAdiate [621386].

References

1. Asamer, J., Reinthaler, M., Ruthmair, M., Straub, M., Puchinger, J.: Optimizing charging station locations for urban taxi providers. Transp. Res. A **85**, 233–246 (2016)
2. Chung, S.H., Kwon, C.: Multi-period planning for electric car charging station locations: a case of korean expressways. Eur. J. Oper. Res. **242**(2), 677–687 (2015)
3. Czimmermann, P., Kohání, M., Buzna, Ľ.: The design of charging infrastructure for electric vehicles and its properties. In: Proceedings of the 14th International Symposium on Operational Research, pp. 123–128 (2017)
4. Kohání, M., Czimmermann, P., Váňa, M., Cebecauer, M., Buzna, Ľ.: Designing charging infrastructure for a fleet of electric vehicles operating in Large Urban areas. In: Proceedings of the 6th International Conference on Operations Research and Enterprise Systems-ICORES, vol. 1, pp. 360–368 (2017)

5. Kohání, M., Czimmermann, P., Váňa, M., Cebecauer, M., Buzna, Ľ.: Location-scheduling optimization problem to design private charging infrastructure for electric vehicles. In: Parlier, G.H., Liberatore, F., Demange, M. (eds.) Operations Research and Enterprise Systems, pp. 151–169. Springer International, Cham (2018)
6. MirHassani, S.A., Ebrazi, R.: A flexible reformulation of the refueling station location problem. Transp. Sci. **47**(4), 617–628 (2013)

Minimum Dominating Set and Maximum Independent Set for Evaluation of EU Funding Polices in Collaboration Networks

Valentin Bouquet, Kymble Christophe, François Delbot, Gaétan Le Chat, and Jean-François Pradat-Peyre

The pluriannual Framework Programme (FP) created in 1984 quickly became the main instrument used by the European Community (EC) to regulate, coordinate and support Research and Innovation (R&I) in Europe. Underlying European intervention is that cooperation and network collaboration are factors of socio-economic betterments. FP embodies a large commissioned process of scientific and economic gaps reduction among state-members, while fostering the competitivity of European Union (EU) firms stifled by the American and Japanese competitions [1, 2]. The promoted cooperation in R&I governance underlies that the chaining of national R&I capacity should largely contribute to strengthening the EU innovation system by synchronizing innovation efforts while avoiding cost duplications. With FP, multiple assumptions are made:

1. collaborative research is more effective than single investigator research [8];
2. projects involving heterogeneous entities (University, Research Institute, firm, association or public administration) are more likely to succeed;
3. likewise for projects with actors located in different regions or countries [7].
4. multi-, pluri- and inter-disciplinaries are key to achieve research projects with European scope.

V. Bouquet · F. Delbot (✉) · J.-F. Pradat-Peyre
Université Paris-Nanterre, Nanterre, France
e-mail: francois.delbot@gmail.com

K. Christophe
Université Paris-Nanterre, Nanterre, France

Economix UMR 7235, Nanterre, France

FRS Consulting, Paris, France

G. Le Chat
FRS Consulting, Paris, France

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_31

All those are gathered within FP7 and are required in order to compete for EU support. Applicants must create formal consortia and mobilize private and public organizations from at least three different nationalities and statutes. Projects must be planed and depicted as optimal responses to specific EU R&I challenges. If so, subsidies decision will be made based on partners and projects qualities and attended socio-economic and environmental output assessed by EC mandated experts. FP partnerships are assumed to ensure secured and innovation conducive environment as identified by Foray [4], it creates temporary zones through which technics, information and knowledges are circulating trustfully. The large scale of FP allows to integer organizations from EU peripheries (which are identified as having relatively few connections with EU core) into the European Research Area, while tightening connections among countries. This is supposed to guarantee a kind of "pop-up" phenomenon whereby EU periphery will economically and scientifically catch-up the core. This ends up in an EU wide collaborative network, that we need to address in order to determine whether it owns properties are compatible with innovation and growth. The aim is to determine whether collaborative networking as promoted by the FP contribute to EU R&I policy achievements. For this purpose, network analysis remains the privileged tool as it allows to appreciate the adequation between EU goals and network structure. There are numerous articles questioning EU FP network structure [5] and trying to identify central agents; they mobilize basic tools as centrality index; but none of them use specific tools as Minimum Dominating Set (MDS) or Maximum Independent Set (MIS) as proposed in this article. We propose to clearly position EU actors on EU innovation skeleton. This will allow us to identify EU innovation backbone and characterized key actors of EU FP innovation system. Since this approach exceed basic observation of budget or projects allocation among countries or entities, it seems more robust by calling for fundamental network structure tools. As said before, the principal goal underlying the creation of innovation programs is the reach of a homogenous and integrated European Research Area materialized by an EU collaborative network backbone composed in half by EU peripheric countries or organizations.

Material and Methods In this paper, we study projects supported by the pluri-annual Framework Programs FP7 and H2020. The data were downloaded on July 5, 2018 from the Community Research and Development Information Service (CORDIS) website [3]. We cleaned up collected data and improved their quality by deleting records for which crucial data was missing (country, id, projectRcn or projectID). The number of deleted records for FP7 was 718 on a total of 144,591 and only 2 for H2020 on a total 76,811. It is important to note that the difference in the number of records is due to the fact that H2020 is still ongoing. From these data, we generated two types of graphs for both FP7 and H2020 (simple, without loops and non-oriented), i.e. a total of four graphs. The first type of graph, also called countries graph, correspond to the relations between the countries of the different organizations. The vertices of the graph correspond to the different countries in which at least one organization is involved in at least one of the projects founded by FP. Two countries A and B are connected by an edge (A, B)

if two organizations are implied in the same project, the first located in A and the second in B. The second type of graph, also called organization's graph, correspond to the relationships between the organizations involved in the different projects. Each vertex corresponds to an organization and two organizations are connected by an edge if they are involved in the same project founded by FP. From these two graphs (organization FP7 and organization H2020), we also generated the intersection graph, containing the vertices present in both FP7 and H2020. In this graph, an edge (A, B) exists if this same edge exists either in organization FP7 or in organization H2020. Each organization's graph contain a giant connected component. We focused our work on it. Definitions can be found in [6] (Tables 1, 2, 3 and 4).

Some structures of the graph's theory such as the minimum dominating set can be used to determine which members are most involved in these collaborative networks. More precisely, such a structure can be seen as the core of the network. And it's different organizations could have an important role in disseminating the knowledge generated by the different projects.

Definition 1 (MDS) Let $G = (V, E)$ be an undirected, unweighted graph (connected or not), with V the set of vertices and E the set of edges. A dominating set $S \subseteq V$ of G is a set of vertices such that $\forall v \in V - S,\ N(v) \cap S \neq \emptyset$, with

Table 1 Properties of countries graphs for FP7 and H2020

	FP7	H2020
Coutries	178	150
Connected	Yes	Yes
Edges	4874	3629
Minimum degree	3	4
Average degree	54	48
Maximum degree	165	138
Density	0.309	0.324
Small world	Yes	Yes
Global clustering coeff.	0.6	0.6
Average distance	1.7	1.7
Diameter	3	3

Table 2 Countries with highest degree

FP7	H2020
FR:168	UK:134
UK:162	DE:133
DE:158	IT:127
IT:158	FR:124
ES:149	ES:123
BE:140	BE:118
PT:136	NL:117

Table 3 Properties of organization's graphs for FP7 and H2020

	FP7	H2020	Inter
Organizations	30,438	23,106	9596
Components	229	2706	2513
Edges	752,112	427,114	98,393
Minimum degree	2	2	2
Average degree	49	36	20
Maximum degree	7496	4218	1758
Density	0.0016	0.0016	0.0021

Table 4 Properties of the giant component of organization's graphs

	FP7	H2020	Inter
Organizations	30,175	20,116	7005
Edges	751,788	423,690	95,707
Minimum degree	3	3	3
Average degree	49	42	27
Maximum degree	7496	4218	1758
Density	0.0016	0.0021	0.0039
Small world	Yes	Yes	Yes
Global clustering coeff.	0.12	0.144	0.237
Average distance	2.78	2.85	2.98
Diameter	6	7	9

$N(v) = \{u : (u, v) \in E\}$. A minimum dominating set (MDS) is a dominating set of minimum size.

In the same way, a maximum independent set represents organizations that do not collaborate together. It might be interesting to change the rules of the FP in order to reduce it's size, and then increase collaborations.

Definition 2 (MIS) An independent set $I \subseteq V$ of G is a set of vertices such that no two vertices in the subset are linked by an edge. A maximum independent set (MIS) is an independent set of maximum size.

Since both countries graphs are of reasonable size, we were able to enumerate all the solutions for MDS and MIS for both graphs. We have noticed that for MDS some countries, like the United Kingdom for H2020, are present in all solutions, which brings us to propose the following definition:

Definition 3 (Persistence) Given a problem P, such that a solution to this problem consists of a collection of discrete elements, an element r is persistent if r is present in each solution of P.

We sought to determine the set of persistent vertices for the MDS and the MIS in organization's graphs. To determine if a vertex is persistent, we use the following method. First, we calculate an optimal solution for MDS (we model the problem as a linear program, then we calculate an optimal solution with the PuLP solver). If a vertex is persistent, then it is necessarily part of that solution. Then, we consider

each vertex of this solution one by one. We remove it from the graph and we calculate again an optimal solution. If the size of the optimal solution has increased for the MDS, then this implies that this vertex is persistent. This method is feasible because the graph is small world, low density, and contains many cliques. We used a similar method for MIS. We performed these calculations in parallel on a server (x86_64, 24 CPUs at 2659.823 MHz and 148Go of memory) and it takes about 1 day to perform the calculation.

Results (1) There is only one solution of size 3 for the MDS on the countries graph for FP7 (Ghana, Fance and Italy). (2) There are 23 distinct optimal solutions of size 4 for the MDS on the countries graph for H2020. United Kingdom is present in all 23 solutions, Italy in 22 solutions, Switzerland in 19 solutions, Senegal in 5 solutions and Tunisia in 2 solutions. All other countries appear only once. (3) The size of an MDS for the giant connected component of the organization's graph for FP7 is 580 (of which 271 are persistent), 566 for H2020 (of which 245 are persistent) and 723 for inter (of which 286 are persistent). (4) We didn't found any persistent vertex for the MIS on any of the organization's graphs. This is certainly because each project results in a clique, which allows one vertex to be replaced by another in the solution.

Discussion In this section, we discuss the previous results with an economic perspective. Based on the three networks analysis generated from our database (for FP7, H2020 and the intersection of FP7 and H2020), we make an attempt at determining the characteristics explaining the propensity for organisations to be persistent as presented in Table 5. As a consequence, we will establish if EU innovation network respects policy objectives as settled in the introduction. We implement Probit regressions to determine the propensity for organisations to be persistent in FP7, H2020 and both. To do so, we create the binary *persistant8i,j* variable that take one if organisations j are persistent in i = {FP7, H2020, Both} and 0 otherwise. $P(Persistant_{i,j} = 1|X) = \Phi(X'\beta)$ with X = (pluridisciplinary$_{i,j}$, University$_{i,j}$, private for profit$_{i,j}$, rich10$_{i,j}$, richUE15$_{i,j}$, EU contribution$_{i,j}$, participation degree$_{i,j}$, number of projects$_{i,j}$). We integer specific participation characteristics such as whether organisations took part in pluri-disciplinary projects (*pluridisciplinary*); organization's typology (University, Private-for-Profit company, Public administration, Research institutes or others); whether or not organizations are located in one of the 15 richest countries of the EU *(richUE15)* or in one of the 10 richest countries of the world (*rich10*).

We also take into account participation characteristics such as the total amount of subsidies received by the organization*(EU contribution)*, total budget of EU projects (*total budget*) plus *number of projects* organizations were involved in or *participation degree*. Basically, we notice that the propensity to be persistent is positively correlated with the participation characteristics: *number of projects* organisations were evolved in, *participation degree* or *(EU contribution)*. Moreover, persistent nodes were mostly *Universities* taking part in *pluridisciplinary* projects. In regard with nodes localization, we see clear evidences that the 15 EU richest countries dominate FP7 backbone. These findings seem to be the very materialization of EU

Table 5 The propensity to be persistent in FP7, H2020 and inter networks: a probit analysis

Variables	FP7	H2020	Inter
Pluri - disciplinarity	0.749*** (0.0954)	1.349*** (1.349)	1*** (0)
University	0.314** (0.1328)	−0.047 (0.094)	0.803** (0.335)
Private for profit company	0.196 (0.1328)	−0.383*** (0.0849)	0.434 (0.342)
RichUE15	0.1585* (0.0918)	−0.0786 (−0.0786)	0.382** (0.175)
Rich10	−0.388 (0.080)	0.316 (0.03161)	−0.161 (0.1277)
EU contribution	$-1.06e^{-8}$ ($6.38e^{-9}$)	$-9.8e^{-9}$ ($6.31e^{-9}$)	$-2.02e^{-8}$*** ($3.43e^{-9}$)
Total budget	$-1.44e^{-8}$ ($3.14e^{-10}$)	$-1.63e^{-9}$*** ($4.60e^{-10}$)	$-2.03e^{-11}$ ($1.91e^{-10}$)
Number of project	0.009* (0.005)	0.017** (0.0089)	0.176*** (0.0025)
Participation degree	0.0029*** (0.0004)	0.003*** (0.0008)	
Cons	−3.52*** (0.1463)	−3.19*** (0.206)	−3.594*** (0.3499)
Observation	29, 869	23, 106	6412

***0.01; **0.5; *0.1; ( ) standard errors

innovation policy to support Research Excellence in FP7. In fact, Top Research Facilities are involved mainly in pluridisciplinary scientific questions. They are also affiliated with Universities localized in EU economic core countries.

Analysis from H2020 reach more nuanced results. We see that*Universities* are no more dominant in persistent nodes. This is relatively concordant with the 2014 policy reorientation to support more applicative and marketable innovation projects. We note that private-for-profit firms are not yet core in EU innovation networks but there is a slight tendency to support less fundamental research project with more and more concerns for pluridisciplinarity. Countries heterogeneity is no more significative; this lead to conclude that inequality among richest and peripheric state-members is potentially less pronounced in H2020 than in FP7. However, based on inter-FP7-H2020 networks the previous assertion must be nuanced, because the 101 organizations involved in both FP7 and H2020 programs were University localized in one of 15 EU richest countries. We see that despite strong effort to support an economic and scientific catch-up process between peripheric and core countries in EU, the community is not yet homogenous or fully integrated. Strong efforts are still necessary to correct for the dominancy of EU richest countries in the FP collaborative network.

Perspectives This work opens many perspectives. First, from a theoretical point of view. What are the persistent vertices and what are the sufficient and necessary conditions that characterize them? This work is already in progress. From a practical point of view, our different results show UK's involvement in H2020. We began to study the impact of an event like the Brexit using different approaches: (1) Removing a persistent vertex from the network of collaborations, (2) removing all organizations of one country's organizations,we model the problem as a linear program, then we calculate an optimal solution thanks to the PULP solver. (3) removing all projects involving this country.

References

1. Breschi, S., Malerba, F.: Clusters, Networks and Innovation. Oxford University Press, Oxford (2005)
2. Commission of the European Communities: 1980–1990: a new development on the European scientific policy, number 7121 (1982)
3. Community Research and Development Information Service. In: FP7 and H2020 Projects. https://data.europa.eu/euodp/en/data/dataset/
4. Foray: The patent system and the dynamics of innovation in Europe. Sci. Public Policy **31**, 449–456 (2004)
5. Malerba, F., Vonortas, N., Breschi, S., Cassi, L.: Evaluation of progress towards a European research area for information society technologies. In: Report to European Commission, DG Information Society and Media (2006)
6. Newman, M.E.J.: The structure and function of complex networks. SIAM Rev. **45**, 167–256 (2003)
7. Presser, S.: Collaboration and the quality of research. Soc. Stud. Sci. **10**, 95–101 (1980)
8. Sonnenwald, D.H.: Scientific collaboration. In: Annual Review of Information Science and Technology (2007)

PHOEG Helps to Obtain Extremal Graphs

Gauvain Devillez, Pierre Hauweele, and Hadrien Mélot

1 Introduction

A generic problem in Extremal Graph Theory consists in finding bounds on some graph invariants with respect to some constraints. A graph invariant is a value (usually numerical) that is preserved by isomorphism. Some common invariants include the order and size of the graph, the average distance, chromatic number, etc. The solutions to these problems are bounds (often parameterized by the value of some other invariant in the constraints) and the graphs realizing those bounds. Indeed, such extremal graphs are proofs that the bounds are tight.

These solutions obviously need to be true for all graphs respecting the given constraints and these graphs can be numerous. An often used constraint is to fix the order n of the graphs. But even so, there are already more than a billion of graphs with 12 vertices. This huge quantity of data creates a need for techniques to determine the extremal graphs and also to help to prove their extremality.

The first project to provide these helps, called Graph, was done by Cvetkovic et al. [6]. This later led up to a new version called newGRAPH by Brankov et al. [3]. But this tool was only the first of a kind and many other tools were developed. In 1988, Graffiti was developed by Fajtlowicz [9] and, using heuristics and pre-computed data, was able to generate more than 7.000 conjectures in its first execution. Later, in 2000, Caporossi and Hansen developed AutoGraphiX [5] which uses the variable neighborhood search metaheuristic to determine good candidates for the extremal graphs. Digenes [1] (2013) uses genetic algorithms and provides support for directed graphs. In 2008, Mélot presented GraPHedron [12], using an exact approach on small graphs. Its central idea is to use all the graphs up to some

G. Devillez · P. Hauweele (✉) · H. Mélot
Algorithms Lab, University of Mons, Mons, Belgium
e-mail: pierre.hauweele@umons.ac.be

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_32

order in the invariant space and then compute the convex hull of these points. The facets of the hull can be seen as inequalities between the chosen invariants and the vertices of the convex hull as extremal graphs.

While tools such as AutoGraphiX and Graffiti have evolved over the years [4, 7], GraPHedron did not. This is why we started a complete overhaul of this tool. This successor, PHOEG,[1] contains a set of tools aimed at speeding up the testing of ideas and helping raise new ones. It is mainly composed of a database of graphs enabling fast queries and computations but also of a module named TransProof whose goal is to assist finding proofs for the conjectures. PHOEG is already being used in scientific research and has led to several results [8, 10].

In the following sections, we present the different aspects of PHOEG and its new features with respect to GraPHedron and explain some of the main ideas used to help the researcher in studying Extremal Graph Theory.

2 Notations and Definitions

Common Graph Theory concepts and notations will be used. However, we define here some specific notions and notations used in this paper.

In our work, we consider only undirected simple graphs. We note $G \simeq H$ if the two graphs G and H are isomorphic. In the computations, we only use one representative for each isomorphism class called the canonical form.

Section 4 explains how we use the convex hull of a set of points corresponding to graphs in order to produce conjectures.

Definition 1 Given a finite set of points $S = x_0, x_1, \ldots, x_n$ in a p-dimensional space, the *convex hull* of S (denoted $conv(S)$) is the intersection of all the convex sets containing S. This set forms a convex polytope.

This polytope can be represented as an intersection of halfspaces. It can thus be represented as a system of linear inequalities.

Definition 2 A *face* of the polytope $\mathcal{P}$ is defined as the intersection of $\mathcal{P}$ with a tangent hyperplane. If $\mathcal{P}$ is of dimension p, a $(p-1)$-dimensional face is called a facet.

In Sect. 6, we use graph transformations to help prove the generated conjectures.

Definition 3 We define a *parameterized graph transformation* $\tau(G, V, E)$ as a function that, given a graph G and a set of edges and vertices of G (resp. E and V), produces a new graph by modifying the given elements of G. e.g., $addedge(G, \{a, b\}, \{\})$ will produce the graph obtained from G by adding the edge ab.

[1] PHOEG stands for PHOEG Helps to Obtain Extremal Graphs.

Definition 4 Given a parameterized graph transformation $\tau(G, V, E)$, a *graph transformation* $\tau(G)$ is defined as the function outputting the results of the application of $\tau(G, V, E)$ to G for every possible V and E. e.g., from $addedge(G, \{a, b\}, \{\})$, we can define $addedge(G)$ as the graph transformation outputting all the graphs that can be obtained from G by adding an edge.

3 Invariants Database

As defined in Sect. 1, a graph invariant is a value that is preserved by isomorphism. While it is fast and easy to compute some invariants on some classes of graphs, others can be computationally costly. This is especially inconvenient when there are millions of graphs to consider. This problem was tackled in GraPHedron by only computing their values once for each graph and then storing their values in files.

As invariants are constant by isomorphism, each graph is only considered once thanks to its canonical form computed by the nauty software [11].

In PHOEG, the data storage and query answering are delegated to a relational database management system. Complete finite classes of graphs, alongside their invariants values, are listed in tables, e.g., all non-isomorphic graphs, trees, claw-free connected graphs, up to some order.

The goal of this tool is to be used by researchers in Graph Theory. As they are not necessarily accustomed to the writing of SQL queries, the addition of a domain specific language for those kinds of queries is a planned feature.

4 Invariants Space

In Extremal Graph Theory, many results are expressed as inequalities between graph invariants [2, 9]. This observation was used by GraPHedron by converting graphs to points whose coordinates are the values of invariants. The convex hull of these points is then computed.

The facets of this convex hull provide inequalities between the invariants used as coordinates. These correspond to bounds on their values that can be of use to define Extremal Graph Theory conjectures. We note that this idea has led to several results [12].

Figure 1 shows a visual example of this process for the eccentric connectivity index (ξ^c).

Definition 5 Given a graph G, the *eccentric connectivity index*, denoted by $\xi^c(G)$ is a graph invariant defined as the sum for all vertices v of G of the product between its degree and its eccentricity. i.e., $\xi^c(G) = \sum_{v \in V(G)} d(v)ecc(v)$.

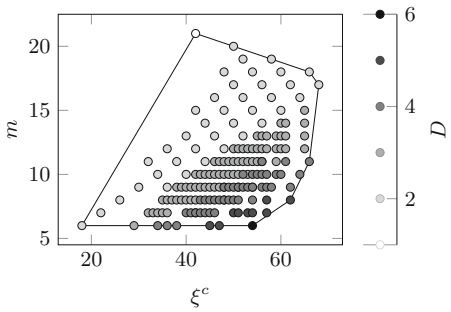


Fig. 1 Polytope and graph coordinates for the eccentric connectivity index (ξ^c) invariant with fixed order ($n = 7$) and parameterized size

The points correspond to the connected graphs of order seven in the invariants space (the size m and ξ^c) and the colour of each point represents the maximum diameter of the graphs with same coordinates.

One of the specificities of PHOEG is that it is possible to explore the inner points of the polytope. For example, GraPHedron did not have the possibility to obtain the diameter of the graphs inside the convex hull.

5 Forbidden Graph Characterization

In Graph Theory, classes of graphs are often described by means of a forbidden graph characterization. Such a characterization of a class $\mathcal{G}$ of graphs is given by an obstruction set $\mathcal{O}$ containing the forbidden graphs. A graph G is a member of $\mathcal{G}$ if and only if it has no element of $\mathcal{O}$ as substructure (e.g., induced subgraph, graph minor). A new feature in PHOEG provides a tool addressing this matter. The substructure relations define a preorder. Given a finite class of graphs or a class membership function and a specific substructure relation, PHOEG computes the minimal graphs (not) in the class for this relation. These minimal graphs are obtained by folding over the class of graphs into an accumulated list of candidate minimal graphs which can be ruled out later during the iteration. If the iteration of the class satisfies the natural order relation of the couple (order, size) of the graphs, then the graphs added to the accumulator are actual minimals and therefore do not need to be ruled out. The output set of minimal graphs provides a proposed obstruction set for the forbidden graph characterization of the input class.

6 Proofs by Transformations

After obtaining conjectures, one needs to prove them, and PHOEG proposes parts of the answer to the problem. Let $\mathcal{G}$ be the set of graphs concerned by the conjecture and $\mathcal{E} \subseteq \mathcal{G}$, the set of extremal graphs, a common technique is a proof

by transformation. These proofs work by defining a set of graph transformations (denoted by $\mathcal{T}$) such that, for any graph $G \in \mathcal{G} \backslash \mathcal{E}$, there is a transformation returning a new graph $H \in \mathcal{G}$, whose value for the studied invariant is closer to the conjectured bound.

One of the most difficult parts of such proofs is to find good transformations. Actually, one not only wants correct transformations but also wants simple transformations (transformations with little impact on the graph they are applied to) to simplify the proof and as few as possible to avoid long and repetitive proofs.

These proofs can also be seen as a directed graph. The vertices of this graph are the graphs concerned by the conjecture ($\mathcal{G}$) and an arc from vertex A to vertex B means that there is a transformation from the graph A to the graph B. We call this graph, the *metagraph of transformations*. With this point of view, a proof by transformation is correct if the metagraph is acyclic and all its sinks (vertices with no exiting arcs) are extremal graphs.

Definition 6 Let $\mathcal{G}$ be a set of graphs and $\mathcal{T}$ be a set of graph transformations. Let M be the directed graph with vertex set $\mathcal{G}$ and arc set $\{(G, U) \in \mathcal{G} \times \mathcal{G} | \exists \tau \in \mathcal{T} \wedge \exists H \in \tau(G), H \simeq U\}$. This graph is the *metagraph of transformations* for the given graph set $\mathcal{G}$ and transformation set $\mathcal{T}$. An example is given in Fig. 2.

This idea is exploited in the TransProof module. The metagraph is pre-computed, for a given set of graphs and transformations, and stored inside a database using the graph data model.

This provides a basis on which tools can be built to allow study of the metagraph. It can be used to test the efficiency of a transformation if used in a proof but also, to help refine them if they are not correct. This mechanism can also be exploited in building heuristic tools to provide good candidates for such proof by making the evaluation of a transformation faster than recomputing it every time.

However, as the number of potential arcs can be quadratic in the number of graphs, the database grows in size exponentially making it impossible to compute results for all graphs and even less for all transformations. This brings need to resort to heuristics in order to help construct proofs by transformation.

In order to reduce the size of the metagraph, one can filter symmetries of a given graph. As an example, one can consider adding an edge to a cycle. There is a lot of ways to do it but many of them are similar. To avoid generating

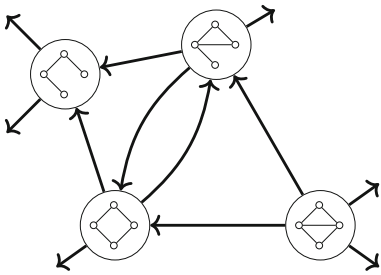


Fig. 2 Metagraph example for edge deletion and rotation

these symmetrical transformations, we use nauty in order to compute the orbits of the automorphism group. i.e., the sets of vertices which can be swapped without changing the representation of the graph. These orbits can then be used to identify the symmetries we are trying to filter out.

Another idea to try and overcome these limitations is that a transformation can be described as a sequence of simple transformations. For example, a commonly used transformation is the rotation.

Definition 7 Let $G = (V, E)$ be a graph. Let a, b, c be vertices in V such that $ab \in E$ and $ac \notin E$. The *rotation* on G using the vertices a, b, c is a parameterized graph transformation that removes edge ab and adds edge ac.

A rotation thus consists in removing an edge and adding another one sharing exactly one extremity. Both these transformations are really simple ones. We could then only define some simple transformations to be precomputed and stored in the database and use them to generate other more complex transformations by chaining them and adding constraints.

The choice of this basis of transformations requires to be able to generate all transformations from a subset of the simple ones. As an example, one can consider increasing the number of connected components of a complete graph. This requires removing a number of edges depending on the graph the transformation is applied to.

This means that we need a way to specify how a variable number of edges will be added or removed. We should thus add different transformations based on the ways one can connect or disconnect a subset of vertices of a graph.

With this basis of transformations, we need only to compute simple transformations and store them inside the graph database. This data can then be exploited by queries to the database for more complex transformations. To this extent, a specific language is being developed where a statement consists of a list of transformations to apply to some parts of the graph and, for each of these transformations, a set of constraints potentially empty. We can thus consider more complex transformations but, for the same reasons as evoked in Sect. 3, we are still limited to small graphs.

At the time of writing, the graph database contains several simple transformations such as removing, adding or rotating an edge for all graphs up to order nine. This data is currently being used to assist finding proofs for several conjectures.

7 Conclusion and Future Work

We explained how PHOEG exploits its databases to help find conjectures in Extremal Graph Theory but also characterize classes of graphs via forbidden subgraphs and construct proofs by transformations. PHOEG is already used for some open problems in Extremal Graph Theory but there is always room for improvement. Some features we are currently working are, setup of a web-service with a Domain Specific Language for queries, heuristic methods to infer polytopes on higher orders

of graphs, techniques to extract common structures in transformations to give more insight on their effects on some invariant.

References

1. Absil, R., Mélot, H.: Digenes: genetic algorithms to discover conjectures about directed and undirected graphs (2013). ArXiv preprint:1304.7993
2. Aouchiche, M.: Comparaison automatisée d'invariants en théorie des graphes. École polytechnique, Palaiseau (2006)
3. Brankov, V., Stevanovic, D.: An invitation to newgraph. In: Rendiconti del Seminario Matematico di Messina, Serie II, vol. 25, 211–216 (2003)
4. Caporossi, G.: Variable neighborhood search for extremal vertices: the autographix-III system. Comput. Oper. Res. **78**, 431–438 (2017)
5. Caporossi, G., Hansen, P.: Variable neighborhood search for extremal graphs: 1 the autographix system. Discret. Math. **212**(1), 29–44 (2000)
6. Cvetkovic, D., Kraus, L., Simic, S.: Discussing Graph Theory with a Computer I. Implementation of Graph Theoretic Algorithms. University of Belgrade, Serbia (1981)
7. DeLaVina, E.: Graffit. PC. Graph Theory Notes of New York **42**(3), 26–30 (2002)
8. Devillez, G., Hertz, A., Mélot, H., Hauweele, P.: Minimum eccentric connectivity index for graphs with fixed order and fixed number of pending vertices. CoRR **abs/1809.03158** (2018). http://arxiv.org/abs/1809.03158
9. Fajtlowicz, S.: On conjectures of graffiti. Discret. Math. **72**, 113–118 (1988)
10. Hauweele, P., Hertz, A., Mélot, H., Ries, B., Devillez, G.: Maximum eccentric connectivity index for graphs with given diameter. CoRR **abs/1808.10203** (2018). http://arxiv.org/abs/1808.10203
11. McKay, B.D., Piperno, A.: Practical graph isomorphism, {II}. J. Symb. Comput. **60**(0), 94–112 (2014). https://doi.org/10.1016/j.jsc.2013.09.003. http://www.sciencedirect.com/science/article/pii/S0747717113001193
12. Mélot, H.: Facet defining inequalities among graph invariants: the system graphedron. Discret. Appl. Math. **156**(10), 1875–1891 (2008)

The Knapsack Problem with Conflict Graphs and Forcing Graphs of Bounded Clique-Width

Frank Gurski and Carolin Rehs

1 Introduction

The knapsack problem is one of the most famous NP-hard tasks in combinatorial optimization. Within the *knapsack problem (KP)* there is given a set $A = \{a_1, \ldots, a_n\}$ of n items. Every item a_j has a profit p_j and a size s_j. Further there is a capacity c of the knapsack. All values are assumed to be positive integers. The task is to choose a subset A' of A, such that the total profit of A' is maximized and the total size of A' is at most c.

We consider the following two types of constraints. A *conflict constraint* C on a pair of items defines the incompatibility of these items. For each conflicting pair (a_i, a_j), at most one item may occur in a feasible knapsack solution. A *forcing constraint* F on a pair of items defines the necessity of these items. For each forcing pair (a_i, a_j), at least one item must occur in a feasible knapsack solution.

Both of these constraints can be expressed by a graph $G = (V, E)$ on n vertices, such that every vertex $v_j \in V$ corresponds to an item $a_j \in A$ and $\{v_i, v_j\} \in E$ means that a_i and a_j are in a conflict or in a forcing constraint. This leads to two extended knapsack problems.

The *knapsack problem with conflict graph (KCG)* is given an instance for KP and a conflict graph G. The goal is to select the maximum profit set of compatible items while satisfying the knapsack capacity constraint. KCG was first considered by Yamada et al. in [13]. The *knapsack problem with forcing graph (KFG)* is given an instance for KP and a forcing graph. The goal is to select the maximum profit set of compatible items while satisfying the knapsack capacity constraint. KFG was first considered by Pferschy and Schauer in [11].

F. Gurski (✉) · C. Rehs
University of Düsseldorf, Institute of Computer Science, Algorithmics for Hard Problems Group, Düsseldorf, Germany
e-mail: frank.gurski@hhu.de; carolin.rehs@hhu.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_33

Both KCG and KFG are strongly NP-hard by a reduction from the maximum independent set problem. But there are pseudo-polynomial solutions for special conflict graphs and forcing graphs. Their running times depend on the profit sum $P = \sum_{j=1}^{n} p_j$. The knapsack problem with chordal conflict and forcing graphs on n vertices and m edges can be solved in time $\mathcal{O}((n+m)P^2)$, see [10, 11]. The knapsack problem with a tree as conflict graph or forcing graph can be solved in time $\mathcal{O}(nP^2)$ and the knapsack problem with conflict graphs and forcing graphs of tree-width at most k can be solved in time $\mathcal{O}(2^{k+1}nP^2)$, see [10, 11].

Similar to the relation between the minimum weight vertex cover problem and the maximum weight independent set problem, we can find a solution for the maximum knapsack problem with forcing graph by a solution for the minimization knapsack problem with conflict graph. Within the *minimization knapsack problem (MKP)* there is given a set $A = \{a_1, \ldots, a_n\}$ of n items. Every item a_j has a profit p_j and a size s_j and there is a capacity c of the knapsack. The task is to choose a subset A' of A, such that the total profit of A' is minimized and the total size of A' is at least c.

Theorem 1 ([11]) *Every optimal solution A' of profit z for KP using profits p_j, sizes s_j, and capacity c with forcing graph G corresponds to an optimal solution $A - A'$ of profit $P - z$ for MKP using profits p_j, sizes s_j, and capacity $\sum_{j=1}^{n} s_j - c$ with conflict graph G, and vice versa.*

2 Solutions for Co-Graphs

As an introductory example we will show how to solve the maximum knapsack problem with co-graphs as conflict graphs. Co-graphs (short for complement reducible graphs) have been introduced in the 1970s by a number of authors under different notations [2]. From an algorithmic point of view a recursive definition based on the following operations is very useful. Let $G_1 = (V_1, E_1), \ldots, G_k = (V_k, E_k)$ be k vertex-disjoint graphs. The *disjoint union* of $G_1, \ldots, G_k$, denoted by $G_1 \oplus \ldots \oplus G_k$, is the graph with vertex set $V_1 \cup \ldots \cup V_k$ and edge set $E_1 \cup \ldots \cup E_k$. The *join composition* of $G_1, \ldots, G_k$, denoted by $G_1 \otimes \ldots \otimes G_k$, is defined by their disjoint union plus all possible edges between vertices of G_i and G_j for all $1 \leq i, j \leq k, i \neq j$.

Definition 1 (Co-Graphs) The class of *co-graphs* is recursively defined as follows.

(i) Every graph on a single vertex $(\{v\}, \emptyset)$, denoted by v, is a *co-graph*.
(ii) If $G_1, \ldots, G_k$ are vertex-disjoint co-graphs, then the disjoint union $G_1 \oplus \ldots \oplus G_k$ and the join composition $G_1 \otimes \ldots \otimes G_k$ are *co-graphs*.

By this definition every co-graph can be represented by a tree structure, denoted as *co-tree*. The leaves of the co-tree represent the vertices of the graph and the

inner nodes of the co-tree correspond to the operations applied to the subexpressions defined by the subtrees. For every graph G one can decide in linear time, whether G is a co-graph and in the case of a positive answer construct a co-tree for G, see [3]. Using the co-tree a lot of hard problems have been shown to be solvable in polynomial time when restricted to co-graphs [2]. W.l.o.g. we can assume that co-trees are binary trees.

Let $G = (V, E)$ be a conflict co-graph defined by some binary co-expression X and I be some instance of KP. For some subexpression X' of X let $F(X', p)$ be the minimum size of a solution with profit p in the graph defined by X'. In order to solve the KCG problem we traverse co-tree T in a bottom-up order and perform the following computations depending on the type of operation.

1. Let $X = v_j$, $v_j \in V$. We initialize $F(v_j, 0) = 0$ for every $1 \leq j \leq n$ and $F(v_j, p)$ for every $1 \leq j \leq n$ and $1 \leq p \leq P$ by

$$F(v_j, p) = \begin{cases} s_j & \text{if } p = p_j \\ \infty & \text{if } p \neq p_j. \end{cases}$$

2. $F(X_1 \oplus X_2, p) = \min_{p'+p''=p} F(X_1, p') + F(X_2, p'')$ for every $0 \leq p \leq P$
3. $F(X_1 \otimes X_2, p) = \min(F(X_1, p), F(X_2, p))$ for every $0 \leq p \leq P$

There is a solution with profit p for the KCG problem on KP instance I with a co-graph defined by co-expression X as conflict graph if and only if $F(X, p) \leq c$, i.e. $OPT(I) = \max\{p \mid F(X, p) \leq c\}$.

Theorem 2 (★[1]) *The KCG problem with a co-graph on n vertices and m edges as conflict graph can be solved in $\mathcal{O}(nP^2 + m)$ time and $\mathcal{O}(nP)$ space.*

By Theorem 1 we can solve the maximum knapsack problem with forcing graph by every algorithm for solving the minimization knapsack problem with conflict graph. The solution given above can be adapted to solve the minimization knapsack problem with conflict graph.

Theorem 3 *The KFG problem with a co-graph on n vertices and m edges as forcing graph can be solved in $\mathcal{O}(nP^2 + m)$ time and $\mathcal{O}(nP)$ space.*

3 Solutions for Clique-Width Bounded Graphs

Next we generalize the results given for co-graphs by considering graph classes of bounded clique-width. Clique-width measures the difficulty of decomposing a graph into a special tree-structure. The clique-width of graphs has been defined by Courcelle and Olariu in [4] as follows.

[1] The proofs of the results marked with a ★ are omitted due to space restrictions.

Definition 2 (Clique-Width) The *clique-width* of a graph G is the minimum number of labels needed to define G using the following four operations:

1. Creation of a new vertex v with label a (denoted by $a(v)$).
2. Disjoint union of two labeled graphs G and H (denoted by $G \oplus H$).
3. Inserting an edge between every vertex with label a and every vertex with label b ($a \neq b$, denoted by $\eta_{a,b}$).
4. Change label a into label b (denoted by $\rho_{a \to b}$).

An expression is called a k-expression if it uses at most k different labels. By this definition every graph of clique-width at most k can be represented by a tree structure, denoted as *k-expression-tree*. The leaves of the k-expression-tree represent the vertices of the graph and the inner nodes of the k-expression-tree correspond to the operations applied to the subexpressions defined by the subtrees.

For special graph classes we can compute a clique-width expression in linear time. This holds for distance-hereditary graphs [6] and for several graph classes allowing a few number of P_4 [5]. Even for general graphs we can obtain an expression by the result of [9] which computes for every graph G of clique-width k a clique-width $2^{3k+2} - 1$-expression for G in time $\mathcal{O}(n^9 \log n)$.

The set of all graphs of clique-width at most 2 is the set of all co-graphs [4] and every graph of tree-width at most k has clique-width at most $3 \cdot 2^{k-1}$, see [1]. The tree-width can only be bounded by the clique-width under certain conditions [7].

Let $G = (V, E)$ be a conflict graph defined by some clique-width k-expression X and I be some instance of KP. For some subexpression X' of X let $F(X', L, p)$ be the minimum size of a solution U with profit p in the graph defined by X' such that the set of labels of vertices in U is L. In order to solve the KCG problem, we traverse k-expression-tree T in a bottom-up order and perform the following computations depending on the type of operation.

1. Let $X = a(v_j)$, $a \in [k]$, $v_j \in V$. We initialize $F(a(v_j), \emptyset, 0) = 0$ for every $1 \leq j \leq n$ and $F(a(v_j), L, 0) = \infty$ for every $1 \leq j \leq n$ and $L \neq \emptyset$. For every $1 \leq j \leq n$, $L \subseteq [k]$, and $1 \leq p \leq P$ we define

$$F(a(v_j), L, p) = \begin{cases} s_j & \text{if } L = \{a\} \text{ and } p = p_j \\ \infty & \text{otherwise.} \end{cases}$$

2. For a disjoint union operation of X and Y for every $L \subseteq [k]$ and every $0 \leq p \leq P$ we define

$$F(X \oplus Y, L, p) = \min_{L = L_1 \cup L_2, p = p_1 + p_2} F(X, L_1, p_1) + F(Y, L_2, p_2).$$

3. For an edge insertion operation on X for every $L \subseteq [k]$ and every $0 \leq p \leq P$ we define

$$F(\eta_{a,b}(X), L, p) = \begin{cases} F(X, L, p) & \text{if } \{a, b\} \nsubseteq L \\ \infty & \text{if } \{a, b\} \subseteq L. \end{cases}$$

4. For a relabeling operation on X for every $L \subseteq [k]$ and every $0 \leq p \leq P$ we define

$$F(\rho_{a\to b}(X), L, p) = \begin{cases} F(X, L, p) & \text{if } a \notin L \text{ and } b \notin L \\ \min(F(X, L, p), F(X, L \cup \{a\}, p), & \\ F(X, (L \cup \{a\}) - \{b\}, p)) & \text{if } a \notin L \text{ and } b \in L \\ \infty & \text{if } a \in L. \end{cases}$$

There is a solution with profit p for the KCG problem on KP instance I with conflict graph defined by k-expression X if and only if there is some $L \subseteq [k]$ such that $F(X, L, p) \leq c$, i.e. $OPT(I) = \max\{p \mid F(X, L, p) \leq c,\ L \subseteq [k]\}$.

Theorem 4 (★) *The KCG problem with a conflict graph on n vertices given by a clique-width k-expression can be solved in $\mathcal{O}(nP^2 2^{2k})$ time and $\mathcal{O}(nP2^k)$ space.*

By Theorem 1 we can solve the maximum knapsack problem with forcing graph by every algorithm for solving the minimization knapsack problem with conflict graph.

Theorem 5 *The KFG problem with a forcing graph on n vertices given by a clique-width k-expression can be solved in $\mathcal{O}(nP^2 2^{2k})$ time and $\mathcal{O}(nP2^k)$ space.*

4 Approximation Results

The given pseudo-polynomial solutions can be used to obtain fully polynomial approximation schemes (FPTAS) as mentioned in [10, 11] for conflict graphs of bounded tree-width and for chordal conflict graphs. One idea is to use a powerful result of Pruhs and Woeginger [12] on the existence of an FPTAS for *subset selection problems*, which can be defined as follows. Given is a set $X = \{x_1, \ldots, x_n\}$ of n elements such that every element x_j has a positive profit p_j and for every subset $X' \subseteq X$ it can be decided in polynomial time whether X' is a feasible solution. Further for every instance there has to be a feasible solution. The task is to find a feasible solution X' of maximum profit. In [12] it has been shown that for every subset selection problem which can be solved by an algorithm of running time polynomial in n and $\sum_{i=1}^{n} p_i$ there is an FPTAS. KCG and also KFG (see [11,

Section 1]) are subset selection problems and we have shown pseudo-polynomial solutions for conflict and forcing graphs of bounded clique-width.

Theorem 6 *There exist FPTAS for KCG on conflict graphs of bounded clique-width and for KFG on forcing graphs of bounded clique-width.*

5 Conclusions

The presented methods allow us to solve the knapsack problem with conflict graph (KCG) and the knapsack problem with forcing graph (KFG) in pseudo-polynomial time for conflict graphs and forcing graphs of bounded clique-width. Since every graph of bounded tree-width also has bounded clique-width [1] we could generalize the known results for conflict graphs and forcing graphs of bounded tree-width of Pferschy and Schauer [10, 11]. There are several further graph classes of bounded clique-width, e.g. graphs with few P_4s and distance hereditary graphs, see [8]. These classes include dense graphs, i.e. graphs with $\Omega(n^2)$ edges.

The time complexity for KCG and KFG restricted to planar graphs, permutation graphs, and HHD-free graphs remain open.

References

1. Corneil, D., Rotics, U.: On the relationship between clique-width and treewidth. SIAM J. Comput. **4**, 825–847 (2005)
2. Corneil, D., Lerchs, H., Stewart-Burlingham, L.: Complement reducible graphs. Discret. Appl. Math. **3**, 163–174 (1981)
3. Corneil, D., Perl, Y., Stewart, L.: A linear recognition algorithm for cographs. SIAM J. Comput. **14**(4), 926–934 (1985)
4. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discret. Appl. Math. **101**, 77–114 (2000)
5. Courcelle, B., Makowsky, J., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. **33**(2), 125–150 (2000)
6. Golumbic, M., Rotics, U.: On the clique-width of some perfect graph classes. In: Proceedings of Graph-Theoretical Concepts in Computer Science. LNCS, vol. 1665, pp. 135–147. Springer, Heidelberg (1999)
7. Gurski, F., Wanke, E.: The tree-width of clique-width bounded graphs without $K_{n,n}$. In: Proceedings of Graph-Theoretical Concepts in Computer Science. LNCS, vol. 1938, pp. 196–205. Springer, Berlin (2000)
8. Kaminski, M., Lozin, V., Milanic, M.: Recent developments on graphs of bounded clique-width. Discret. Appl. Math. **157**, 2747–2761 (2009)
9. Oum, S., Seymour, P.: Approximating clique-width and branch-width. J. Combin. Theory Ser. B **96**(4), 514–528 (2006)
10. Pferschy, U., Schauer, J.: The knapsack problem with conflict graphs. J. Graph Algorithms Appl. **13**(2), 233–249 (2009)
11. Pferschy, U., Schauer, J.: Approximation of knapsack problems with conflict and forcing graphs. J. Comb. Optim. **33**(4), 1300–1323 (2017)

12. Pruhs, K., Woeginger, G.J.: Approximation schemes for a class of subset selection problems. Theor. Comput. Sci. **382**(2), 151–156 (2007)
13. Yamada, T., Kataoka, S., Watanabe, K.: Heuristic and exact algorithms for the disjunctively constrained knapsack problem. Inf. Process Soc. Jpn. J. **43**, 2864–2870 (2002)

Part VIII
Logistics and Freight Transportation

Heterogeneity of Items in an Integrated Item-Sharing and Crowdshipping Setting

Moritz Behrend and Frank Meisel

1 Introduction

An item-sharing platform provides its members a temporary access to items such as tools or leisure equipment so that the members can use the items for their own purposes. Accessing items for the time needed may allow for a more resource-efficient consumption as multiple members can sequentially use a same item instead of each buying it individually. The sharing of items does, however, necessitate a frequent peer-to-peer exchange that poses challenges from a transportation perspective.

To address these challenges, we investigated in [1] the idea of an integrated platform that combines item-sharing with crowdshipping. Crowdshipping means that private drivers receive financial incentives to execute delivery jobs along their intended trips [2]. It has been shown that such an integration is a promising approach to enhance the profitability of a platform and to provide a higher level of service to its members as the two concepts are mutually beneficial. Item-sharing benefits from crowdshipping as it provides the opportunity to efficiently transfer single items between (distant) locations. Conversely, crowdshipping benefits from item-sharing as the delivery jobs are an outcome of a preceding assignment and therefore can be 'fitted' to the crowdshipping trips.

So far, the potential of an integrated platform has only been evaluated in settings with homogeneous items. This provides maximum flexibility to the platform for

The online version of this chapter (https://doi.org/10.1007/978-3-030-18500-8_34) contains supplementary material, which is available to authorized users.

M. Behrend (✉) · F. Meisel
Kiel University, Kiel, Germany
e-mail: moritz.behrend@bwl.uni-kiel.de; meisel@bwl.uni-kiel.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_34

assigning supplied items to requests. In the following, we want to analyze how these results are effected when heterogeneous items are considered. For this purpose, we describe the problem in Sect. 2, we briefly review the solution methods of [1] in Sect. 3, and we present computational results in Sect. 4.

2 Problem Description

2.1 Platform

We consider an integrated platform on which members announce their supplies of items, their requests for items, as well as trips on which they accept making deliveries. The platform collects all announcements over one period and then coordinates the matching of compatible supplies with requests followed by assigning consumers or crowdshippers the task to execute the resulting delivery jobs. Eventually, it responds to the members' announcements at the beginning of the next period. The platform receives a revenue from satisfying requests and it needs to compensate crowdshippers for their transportation services. It's objective is to maximize profit.

2.2 Transfer Modes

We consider three alternative modes to transfer items from one location to another, namely self-sourcing, home delivery, and neighborhood delivery. When a request is satisfied through *self-sourcing* it means that the consumer behind the request picks up the assigned item himself/herself. To this end, he/she performs a round trip from the requesting location to the supply location, and back. The overall travel time a consumer is willing to spend on self-sourcing an item is limited. If this limit is exceeded, a request cannot be satisfied with the assigned item as the associated consumer would refuse to self-source it. We refer to this limit as self-sourcing flexibility.

In a *home delivery*, a supplied item is picked up by a crowdshipper and delivered directly to the designated consumer. The corresponding detour is not allowed to exceed a preset time limit, the so-called detour flexibility. A *neighborhood delivery* describes a combination of self-sourcing and crowdshipping where the crowdshipper picks up an item and takes it to the destination of the trip. Then, the consumer self-sources it from there. A neighborhood delivery consequently describes a two-stage transfer in which a crowdshipper and a consumer collaboratively coordinate the transport. Here, the detour flexibility and the self-sourcing flexibility have to be respected both.

2.3 Heterogeneous Items

We denote by H the set of types of items that are shared on the platform. Examples for elements in H are certain types of tools, leisure equipment, etc. like power drills and gaming consoles. Let $h_i \in H$ denote the type of a supplied item i and let $h_j \in H$ denote the type of a requested item j. A supply i and a request j are considered compatible if both are of the same type, which is indicated by binary parameter $q_{ij} = 1$ if and only if $h_i = h_j$.

2.4 Example

Figure 1 illustrates two exemplary problem settings (top) and possible solutions thereof (bottom). The examples consist of three supplies (+) and three requests (−). The shape of a supply/request node indicates the type of the associated item. Crowdshipping trips are shown by two shaded rectangles for the origin and the destination of a trip that are linked by an arc pointing in the direction of travel.

With homogeneous items (Fig. 1a), the platform has full flexibility in matching supplies with requests ($q_{ij} = 1 \ \forall i, j$). This allows to satisfy all requests and to align the delivery jobs to the crowdshipping trips so that only minor detours are required to execute them. j_1 is satisfied through self-sourcing, j_2 through a neighborhood delivery, and j_3 through a home delivery. With heterogeneous items (Fig. 1b), the platform is restricted to matching requests with compatible supplies.

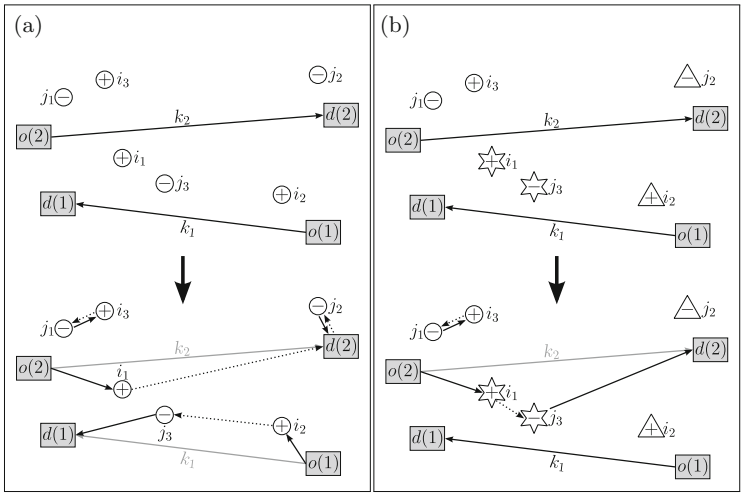


Fig. 1 Illustration of an exemplary problem setting and a possible solution. (**a**) Homogeneous items. (**b**) Heterogeneous items

For example, i_1 is no longer compatible to j_1 and j_2 but only to j_3 ($q_{11} = q_{12} = 0$, $q_{13} = 1$). The platform can consequently assign i_1 only to j_3. When fewer alternative assignments exist, it decreases the extent to which a platform can fit delivery jobs to crowdshipping trips and it results in less efficient transports for crowdshippers. This negatively effects profit and the number of satisfied requests. For example, it is assumed in Fig. 1b that the detour flexibility of both k_1 and k_2 does not suffice to transfer i_2 to j_2. Due to the long travel time between i_2 and j_2, self-sourcing is no option either so that j_2 remains unsatisfied here. Thus, ceteris paribus, a platform with heterogeneous items typically achieves less profit than a platform with homogeneous items.

In the experiments conducted in [1], it was assumed that $q_{ij} = 1 \ \forall i, j$ in all test instances. With this paper, we evaluate the effect of heterogeneous items on the profitability of the platform.

3 Models and Methods

This section briefly reviews the models and methods by [1] that are used to evaluate the benefit of an integrated platform when sharing heterogeneous items. All of them already respect the compatibility parameters q_{ij}, i.e. they support heterogeneous items although this feature has not been evaluated in [1].

We denote by **[ssrc]** the binary program for matching supplies with requests. Every match corresponds to a satisfied request where the consumer behind the request self-sources the assigned item. The model is formulated as an assignment problem that ensures compatibility between supplies and requests, and that respects the limited self-sourcing flexibility of consumers. Crowdshippers cannot be assigned delivery jobs in this model. The objective value of **[ssrc]** therefore serves as a baseline comparison as it represents the maximum achievable profit of a pure item-sharing platform without crowdshipping.

The binary program **[nbrhd]** is an extension of **[ssrc]** in which items can also be crowdshipped. It therefore supports all three transfer modes: self-sourcing, home delivery, and neighborhood delivery. The result is a feasible and profit-maximizing combination of modes to transfer items between supply and request locations. Crowdshippers accept maximum one delivery job per trip which is why the model is formulated as a three-dimensional assignment problem that simultaneously assigns supplies to requests to trips subject to the compatibility between supplies and requests, the self-sourcing flexibility of consumers, and the detour flexibility of crowdshippers. The objective value of **[nbrhd]** corresponds to the maximum achievable profit of an integrated platform with crowdshipping.

To address the trade-off between solution quality and time, [1] also present a graph-theoretical heuristic **[ish]** that seeks for a maximum weighted independent set in an incompatibility graph. A node in the graph corresponds to a valid assignment of a supply to a request and a transfer mode that is weighted with the prospective profit for the platform. Edges between nodes indicate mutual exclusivity as the same

supply, the same request, or the same trip is involved in both incident assignments. A feasible solution is a subset of non-adjacent nodes. The heuristic iteratively selects nodes with a regret-rule, attempting to maximize the total weight of selected nodes.

4 Computational Results

Our computations are based on the instances used in [1], which consider a 30×30 km area centered in the city of Atlanta (Georgia, US). Supply and request locations are distributed within this area according to the population density. Origin and destination locations of trips are extracted from traffic forecasts provided by the Atlanta Regional Commission. It is assumed that drivers travel between locations along the fastest route where routes and travel times are based on the actual street network. We use a 3.4 GHz machine with 4 cores and 16 GB RAM for our computations. The optimization models are solved with Gurobi 7.5.

We evaluate the effect of heterogeneous items based on an experimental set-up with 100 supplies, 100 requests, and 100 trips. Starting from a base setting with one item type (homogeneous), the variety of supplied and requested items is increased to 5, 10, 25, and 50 different types. The frequency of occurrence is identical for all types, e.g. for a setting with five types there are 20 items per type. We further consider scenarios where the number of supplies is higher than the number of requests. The supply-to-request ratio (Sply:Req) is therefore gradually increased from 1:1 to 5:1. The remaining parameters are set as in [1]. Identical revenues are assumed for all types of items.

Table 1 shows the relative improvement of a platform's profit due to an integration with crowdshipping (**[nbrhd]** and **[ish]**) and additional supplies. The reference value for each number of item types is the platform's profit for the setting with 100 supplies and 100 requests, solved under **[ssrc]**. The values reported here refer to average values over 30 random supply-request constellations.

Looking at the results of **[ssrc]** (no crowdshipping) under an increasing number of item types, we see that profits decrease. For example, the profit is \$525 with one type but only \$32 with 50 types. This is because it gets more difficult to satisfy requests the less compatible items are. However, in relative terms, crowdshipping becomes increasingly useful here, as it allows for an efficient transport over long distances so that even remote items can be made accessible to consumers at low costs. For example, integrating crowdshipping increases profits by 21% with one item type but up to 93% with 50 types, see results for **[nbrhd]** in the first row of Table 1.

Furthermore, a platform's profit increases with oversupply, both for a pure item-sharing platform and an integrated one, as well as for a setting with homogeneous items and heterogeneous items. This finding is expected as more opportunities exist to satisfy requests. Note, however, how the supply-to-request ratio needs to increase for a pure item-sharing platform to be as profitable as an integrated platform without oversupply as more types are considered. For the scenario with

Table 1 Computational results (reference values in bold)

Item types	1 (homogeneous)			5			10			25			50		
Sply : Req	[ssrc]	[nbrhd] (%)	[ish] (%)	[ssrc]	[nbrhd] (%)	[ish] (%)	[ssrc]	[nbrhd] (%)	[ish] (%)	[ssrc]	[nbrhd] (%)	[ish] (%)	[ssrc]	[nbrhd] (%)	[ish] (%)
100 : 100	$**525**	21	18	$**206**	45	44	$**126**	56	54	$**58**	81	81	$**32**	93	93
120 : 100	14%	38	34	17%	67	65	18%	82	81	10%	103	101	3%	123	123
140 : 100	25%	48	45	34%	89	86	34%	109	108	23%	124	122	27%	134	134
160 : 100	35%	57	53	56%	113	111	57%	135	133	43%	157	156	57%	190	189
180 : 100	40%	63	59	65%	125	122	65%	150	148	63%	185	183	74%	218	216
200 : 100	46%	69	65	74%	136	133	70%	159	157	75%	209	208	62%	230	229
300 : 100	62%	86	80	134%	201	196	142%	247	243	153%	339	336	137%	372	369
500 : 100	77%	101	96	200%	265	259	245%	354	349	289%	489	483	313%	598	593

homogeneous items, 40 additional items are required (25% vs. 21% profit increase for **[ssrc]** and **[nbrhd]**, respectively). With 50 types, the platform needs more than 100 additional items (62% vs. 93%). This finding emphasizes the benefit of an integration the more diverse the portfolio of offered items gets.

Another relevant performance measure in item-sharing is the share of satisfied requests [3]. To this end, we evaluate the request fulfillment of an integrated platform with heterogeneous items. In case of no oversupply, the platform manages to satisfy 59% of the requests with one type, 18% with 10 types, and 6% with 50 types (numbers not shown in Table 1). With a supply of 500 items, the platform achieves much higher service levels of 95%, 54%, and 21%, respectively. Consequently, both the integration with crowdshipping and additionally supplied items contribute to a significant improvement of consumers' experience with item-sharing.

The independent set heuristic **[ish]** performs very well for all settings. We observe an optimality gap of 1–3% in most cases and all results can be obtained in less than a second. In contrast, **[nbrhd]** requires between 1 and 2 min to solve the largest instances with a supply of 500 items.

5 Conclusions

We have investigated the benefit of integrating item-sharing and crowdshipping on a single sharing platform under homogeneous and heterogeneous items. As expected, our experiments have shown that the profitability of the platform declines the more diverse the set of shared items is. Still, we show that the integration with crowdshipping then becomes even more relevant to capture the spatially spread compatible supplies and requests. Future research should address settings where a crowdshipper cannot delivery just one but multiple items.

References

1. Behrend, M., Meisel, F.: The integration of item-sharing and crowdshipping: Can collaborative consumption be pushed by delivering through the crowd? Transp. Res. B Methodol. **111**, 227–243 (2018)
2. Archetti, C., Savelsbergh, M., Speranza, M.G.: The vehicle routing problem with occasional drivers. Eur. J. Oper. Res. **254**, 472–480 (2016)
3. Lamberton, C.P., Rose, R.L.: When is ours better than mine? A framework for understanding and altering participation in commercial sharing systems. J. Market. **76**, 109–125 (2012)

Metamodel-Based Optimization of the Article-to-Device Assignment and Manpower Allocation Problem in Order Picking Warehouses

Ralf Gössinger, Grigory Pishchulov, and Imre Dobos

1 Problem Description

One usual objective of order picking in warehouses is to fulfill a batch of customer orders within the shortest makespan. Alternative ways to achieve this arise from different ways of combining and utilizing three kinds of resources: articles demanded by customers, devices to pick articles according to customer orders, and operators working at the devices. In this context a *device* is an order picking machine consisting of multiple storage slots, handling equipment, and a limited number of workplaces. In order to pick articles according to customer orders, two types of activities are performed at each device: replenishing storage slots with articles and picking out articles from storage slots. Devices differ in their technical characteristics, in particular pick and replenishment times as well as storage capacity, are different. The devices' capacity can only be utilized, if manpower is allocated to it. Regular fluctuations of *article demand* volume induce a daily sequence of slack and peak periods with known durations. During slack periods, all storage slots of devices are completely replenished so as to reduce the number of replenishments during the subsequent peak period. The assortment of

R. Gössinger (✉)
Department of Production Management and Logistics, University of Dortmund, Dortmund, Germany
e-mail: ralf.goessinger@udo.edu

G. Pishchulov
Alliance Manchester Business School, The University of Manchester, Manchester, UK

St. Petersburg State University, St. Petersburg, Russia

I. Dobos
Institute of Economic Sciences, Budapest University of Technology and Economics, Budapest, Hungary

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_35

articles is subject to daily adjustments. A small number of articles is typically being added and nearly the same number of articles is removed from the assortment. Articles in the assortment differ in their handling requirements so that there may be no single picking technology applicable to all articles. *Operators* differ in their qualification to work at certain devices. There are a number of specialized operators qualified to work at a very small range of devices, and a pool of generalists able to work at a wide range of devices, yet with a lower efficiency. As long as articles are eligible for being picked at multiple devices, an article-to-device assignment has to be determined, and as long as operators are qualified for working at multiple devices, a manpower allocation needs to be specified. Since both decisions have an interdependent influence on the objective value, they form an *article-to-device assignment and manpower allocation* (ADAMA) problem.

Despite of its relevance, the present problem has not yet been extensively discussed in the literature and, to the best of our knowledge, only one solution approach has been proposed which directly addresses the problem [1]. Research on the structurally similar problems of forward-reserve assignment and allocation (FRAAP [2, 3]) as well as of machine loading and manpower allocation (MLMAP [4, 5]) has been reviewed in [1]. In addition, research on *multi-manned assembly line balancing problems* (MALBP) bears resemblance to ADAMA. Here, a set of tasks has to be performed by a set of multi-manned stages with respect to precedence relations and a predetermined cycle time [6]. For the more relevant case of a *heterogeneous workforce*, a MINLP model and a constructive heuristic solution approach is developed in [7]. In order to reduce the solution time with a standard solver, in [8] the formulated MINLP model is approximatively linearized by means of McCormick envelopes (MCE), and a hierarchical solution approach is proposed. Since the approximation error of the linearization is not taken into consideration, the comparison of both approaches is less conclusive.

Regarding the problem discussed in this paper, FRAAP, MLMAP and MALBP are complementary approaches. This paper aims at combining them in order to allow for a more efficient manpower and device utilization. For the purpose of complexity reduction, we adopt a *metamodel-based simulation optimization* approach [9]. A metamodel (MM) is usually generated by running a simulation with complex relations and fitting a simple explicit function that approximately maps observed objective values to combinations of environmental states and alternatives [10]. In contrast to this, we apply a MM for approximating a nonlinear constraint. The MM maps the relation between environmental states and optimal decisions by means of both, MCE [11] and predicted intervals for optimal values of involved decision variables.

The remainder of the paper is organized as follows: In Sect. 2 we derive a decision model for the exact planning approach. Subsequently we develop a MM approach in Sect. 3. At first, the MM is integrated into the exact decision model. After that, prediction models for MM parameters are substantiated and the planning approach is tested with in-sample and out-of-sample instances. Finally, conclusions on the applicability of the approach are drawn in Sect. 4.

2 Exact Planning Approach

Based on the problem description, a simultaneous ADAMA model can be formulated as follows (cf. Table 1 for notations; for a former version cf. [1]):

$\min\ m$			(OBJ)
$d_i \leq \overline{d} \quad \forall i$	(ADA1a)	$\sum_i s_i^f \leq 1$	(MA1a)
$\sum_i a_{ij} = 1 \quad \forall j$	(ADA1b)	$s_i^s \leq 1 \quad \forall i$	(MA1b)
$a_{ij} \leq e_{ij} \quad \forall i,j$	(ADA1c)	$w_i^s \cdot s_i^s + w^f \cdot s_i^f \leq p_i \quad \forall i$	(MA2)
$y_j \cdot a_{ij} \leq (o_{ij} + \rho_{ij}) \cdot c_{ij} \quad \forall i,j$	(ADA2a)	$m \geq d_i \quad \forall i$	(LC1)
$\rho_{ij} \leq M_{ij} \quad \forall i,j$	(ADA2b)	$\rho_{ij} \leq o_{ij} \cdot M_{ij} \quad \forall i,j$	(LC2)
$o_{ij} \leq (y_j \cdot a_{ij} - 1)/c_{ij} + 1 \quad \forall i,j$	(ADA2c)	with $M_{ij} = p_i \cdot \overline{d} / \left(t_{ij}^p \cdot c_{ij} + t_{ij}^r\right) \quad \forall i,j$	
$\sum_j o_{ij} \leq l_i \quad \forall i$	(ADA3)	and $c_{ij} = \lfloor h_i/g_j \rfloor \quad \forall i,j$	
$\sum_j \left(y_j \cdot a_{ij} \cdot t_{ij}^p + \rho_{ij} \cdot t_{ij}^r\right) \leq \left(w_i \cdot s_i^s + \lambda \cdot w^f \cdot s_i^f\right) \cdot d_i \quad \forall i$			(ADAMA1)

The model aims at minimizing the makespan (*OBJ*). The decisions have to pay respect to technical and organizational constraints relevant for an order picking warehouse, which are labeled according to their regard to article-to-device

Table 1 Notations

Indices		w_i^s	number of specialists available for i, with $w_i^s < p_i$
i	device, $i = 1, \ldots, I$	w^f	number of available generalists
j	article, $j = 1, \ldots, J$	y_j	customer demand for j
Parameters		*Decision variables*	
c_{ij}	storage capacity of one slot at i in terms of j	a_{ij}	share of j^{th}demand assigned to i, $a_{ij} \in [0,1]$
$\overline{d}$	duration of the peak period	d_i	total time to fulfill article demand assigned to i, $d_i \in \mathcal{R}_0^+$
e_{ij}	eligibility of j to be picked at i	m	makespan $m \in \mathcal{R}_0^+$
g_j	size of j	o_{ij}	number of storage slots occupied by j at i, $o_{ij} \in \mathcal{N}_0$
h_i	length of one storage slot at i	r_{ij}	number of storage slot replenishments with j at i, $r_{ij} \in \mathcal{R}_0^+$
λ	output ratio of generalists relative to specialists, $0 < \lambda < 1$	ρ_{ij}	number of slot loadings for j at i, with $\rho_{ij} = r_{ij} \cdot o_{ij}$, $\rho_{ij} \in \mathcal{R}_0^+$
l_i	number of storage slots available at i	s_i^f	share of generalists allocated to i, $s_i^f \in [0,1]$
M_{ij}	upper bound for ρ_{ij}	s_i^s	share of specialists allocated to i, $s_i^s \in [0,1]$
t_{ij}^p	time per piece to pick j at i		
t_{ij}^r	time to replenish one slot at i with j		
p_i	number of workplaces at i		

assignment (ADA) or manpower allocation (MA), or both (ADAMA), and their application as linearization constraint (LC). *ADA1*: Demand has to be met within the peak period for each article. That is, (a) the utilization time of each device must not exceed the peak period's duration, and (b) the demand quantity of each article has to be completely allocated to the devices, for which (c) this article is eligible. *ADA2*: Demand for an article allocated to a device is fulfilled with the quantity that is initially stored in occupied slots and the quantities replenished at these slots during the peak period (a). Further, for a specific article at a given device some upper bounds are relevant: The number of slot loadings within the peak period is restricted by the maximum capacity and capacity requirements (b). Furthermore, the allocated demand volume restricts the number of slots that can be occupied (c). *ADA3*: At each device, storage slots can be occupied up to the available number. Each slot can be occupied by only one article, but an article can occupy multiple slots per device. *MA1*: Generalists (a) and specialists (b) can be employed up to 100%. *MA2*: Devices can be manned up to the number of available workplaces. *LC1*: The makespan is the longest time one device needs for fulfilling demand of assigned articles. *LC2*: Since the number of slot loadings is the number of occupied storage slots times the number of slot replenishments, a positive value of the aggregate variable requires a positive number of occupied storage slots. *ADAMA1*: Workload induced by articles assigned to a device has to be met by allocated manpower within the device's utilization time. Multiplication of continuous variables d_i and s_i^s, s_i^f induces a nonlinearity. Hence, ADAMA represents a MIQCP which can be solved with a MINLP solver.

3 Metamodel-Based Planning Approach

One way of facilitating the solution process is to relax the nonlinear right-hand side of ADAMA1. Due to the fact that the terms $d_i \cdot s_i^s$ and $d_i \cdot s_i^f$ are bilinear and that closed intervals for the variables are given ($d_i \in \left[0, \overline{d}\right]$, $s_i^s \in [0,1]$, $s_i^f \in [0,1]$), a linear approximation can be achieved by means of MCE [11]. This provides the formal basis for the *MM*. Consequently, bilinear terms are substituted with aggregate variables $\gamma_i^s, \gamma_i^f \in \mathcal{R}_0^+$ and aggregate variables are constrained by *linear approximations* involving lower d_i^l, $s_i^{s.l}$, $s_i^{f.l}$ and upper bounds d_i^u, $s_i^{s.u}$, $s_i^{f.u}$ of the respective variables:

$\sum_j \left(y_j \cdot a_{ij} \cdot t_{ij}^p + \rho_{ij} \cdot t_{ij}^r\right) \leq w_i \cdot \gamma_i^s + \lambda \cdot w^f \cdot \gamma_i^f \quad \forall i$			(ADAMA1')
$\gamma_i^s \leq s_i^{s.u} \cdot d_i + s_i^s \cdot d_i^l - s_i^{s.u} \cdot d_i^l \ \forall i$	(LC3a)	$\gamma_i^f \leq s_i^{f.u} \cdot d_i + s_i^f \cdot d_i^l - s_i^{f.u} \cdot d_i^l \ \forall i$	(LC3e)
$\gamma_i^s \leq s_i^s \cdot d_i^u + s_i^{s.l} \cdot d_i - s_i^{s.l} \cdot d_i^u \ \forall i$	(LC3b)	$\gamma_i^f \leq s_i^f \cdot d_i^u + s_i^{f.l} \cdot d_i - s_i^{f.l} \cdot d_i^u \ \forall i$	(LC3f)
$\gamma_i^s \geq s_i^{s.l} \cdot d_i + s_i^s \cdot d_i^l - s_i^{s.l} \cdot d_i^l \ \forall i$	(LC3c)	$\gamma_i^f \geq s_i^{f.l} \cdot d_i + s_i^f \cdot d_i^l - s_i^{f.l} \cdot d_i^l \ \forall i$	(LC3g)
$\gamma_i^s \geq s_i^{s.u} \cdot d_i + s_i^s \cdot d_i^u - s_i^{s.u} \cdot d_i^u \ \forall i$	(LC3d)	$\gamma_i^f \geq s_i^{f.u} \cdot d_i + s_i^f \cdot d_i^u - s_i^{f.u} \cdot d_i^u \ \forall i$	(LC3h)

The second basis of the MM ties up to both, possible empirical observations and the property of MCE to provide tighter approximations when the feasibility intervals of involved variables become tightened. A statistical analysis of regularities between the optimal d_i-, s_i^s- and s_i^f-values and environmental states observed for problem instances that have already been solved exactly can be used to *predict intervals of optimal values* for new instances and thus tighten the feasibility intervals. In doing so, on the one hand, two effects will support a more efficient solution process: (1) intervals of optimal values are narrower than intervals of feasible values so that the approximation error can be reduced, and (2) most likely unfavorable values are removed from the solution space in advance so that the solution process can be accelerated. On the other hand, prediction intervals with a low statistical significance are accompanied with the risk of missing the optimal solution or, in extreme cases, rendering the problem infeasible.

In addition, the empirical view allows for estimating intervals of optimal values $[ds_i^{s.l}, ds_i^{s.u}]$, $[ds_i^{f.l}, ds_i^{f.u}]$ for the bilinear terms. Based on this, the co-domains of the aggregate variables can be specified more restrictively: $\gamma_i^s \in \left[ds_i^{s.l}, ds_i^{s.u}\right]$, $\gamma_i^f \in \left[ds_i^{f.l}, ds_i^{f.u}\right]$. Further, in order to reduce the approximation error, the right-hand (left-hand) endpoints of the latter intervals can respectively be used to restrict the right-hand sides in LC3a-b, e-f (LC3c-d, g-h) from above (below).

For the purpose of prediction, a multiple linear regression analysis for fitting *second-order polynomials* that take *two-factor interactions* into consideration is chosen [12]. It is analyzed to what extent the environmental state characteristics have influence on the optimal values of individual and aggregate decisions variables involved in the nonlinear terms of ADAMA1. The environmental state is described by the number of articles J, capacity of storage slots at automated devices C, and fraction of generalists G in the operator pool. Modeling and evaluation [13] refers to 72 optimal instance-level results achieved by attempting to solve the nonlinear ADAMA model for 81 instances with a state-of-the-art MINLP solver (BARON 15.9) on a laptop computer with a 2.4 GHz Intel Core i5 CPU with four cores. The instance generation is based on real and sampled data of a pharmaceutical wholesaler (for details [1]).

The ability of chosen polynomials to predict decisions is evaluated by means of a repeated random sub-sampling cross-validation with 8 validation samples (a to h, each with 9 observations) and 8 training samples (*neg*(a) to *neg*(h), each with 63 observations). In this way 8 out-of-sample evaluation results are available for each regressand, such that the *ex-ante performance* can be examined in terms of determination, significance, accuracy (average prediction error) and robustness (prediction error's coefficient of variation) (cf. Table 2). In sum, it can be concluded that the estimation models fit the data well and are able to predict d (s_i^f, ds_i^f) very well (with acceptable accuracy). On this basis, intervals of optimal values for utilization time, manpower allocation and allocated capacity are determined for the respective maximum values of significance levels from sample mean, unbiased sample variance and regressors' (co-)variances.

Table 2 Performance of polynomials (in all instances: $d = d_i \; \forall \; i, s_i^s = 1 \forall i, \; s_5^f = s_6^f = 0$)

Regressand	$avg(R^2)$	$cv(R^2)$	max(*Sig.*)	min(*Sig.*)	$avg(rMAE)$	$cv(rMAE)$
d	0.992	0.001	0.0001	0.0001	0.0411	0.1846
s_1^f	0.265	0.195	0.0001	0.0001	0.2693	0.4437
s_2^f	0.164	0.183	0.044	0.0001	0.3317	0.4978
s_3^f	0.139	0.207	0.049	0.001	0.3397	0.4023
s_4^f	0.010	0.298	0.089	0.006	0.3446	0.3712
ds_1^f	0.438	0.114	0.0001	0.0001	0.3632	0.2294
ds_2^f	0.450	0.061	0.0001	0.0001	0.4055	0.3153
ds_3^f	0.474	0.044	0.0001	0.0001	0.4032	0.2397
ds_4^f	0.419	0.086	0.0001	0.0001	0.4031	0.1846

Table 3 Indices for solution quality and time of in-sample and out-of-sample instances

	Makespan		Solution time			Makespan		Solution time	
In-sample	*rMAD*	*CVAD*	*rMAD*	*CVAD*	Out-of-sample	*rMAD*	*CVAD*	*rMAD*	*CVAD*
Mean	0.0556	0.7314	0.0024	1.6055	Mean	0.0668	0.7842	0.0025	1.4082
cv	0.0631	0.0790	0.0767	0.0700	cv	0.4682	0.1722	0.4181	0.2069

The MM approach is tested with the samples used for cross-validating the prediction models in such a way that in-sample and out-of-sample results can be compared with the exact approach. The *solution quality* is evaluated in terms of accuracy (relative mean absolute deviation *rMAD* from exact solution) and robustness (absolute deviation's coefficient of variance *CVAD*) of the resulting makespan. Considering that optimal objective values achieved on the basis of MCE underestimate the actual makespans, we derive the makespan values from the ratios of ADAMA1's left-hand side to the bracket term in its right-hand side. The *solution time* is evaluated by the same statistical indices, which indicate the ratio or stability of time savings, respectively. Table 3 summarizes aggregate results for all in-sample and out-of-sample instances.

Solutions to in-sample instances achieved with the MM approach show a slight and relatively stable deviation from the optimum *makespan*. This is also observed for out-of-sample instances, even though the deviation is somewhat bigger and less stable. In comparison to the exact approach, the *solution time* of the MM approach is strongly reduced for in-sample instances. Due to stronger variations, the difference to out-of-sample instances is not significant. Against the background of an empirical basis limited to 63 observations, which are used for estimating sample-related MMs, these results allow for interpreting the performance of the proposed approach to be better than acceptable.

4 Conclusions

For warehouses with heterogeneous resources we propose a combination of an exact and a metamodel-based approach (MM approach) for assigning articles and allocating manpower to devices in an integrative way. Due to the presence of a nonlinear constraint, the exact approach cannot handle real-world problems within the time span between two changes in the assortment of articles but it delivers optimal results when more time is available. The MM is estimated on the basis of these optimal results and relaxes the nonlinear term by means of both, MCE and predicted intervals for optimal values of involved individual and aggregate variables. A numerical study with sampled data of a pharmaceutical wholesaler reveals that this approximate linearization and concentration on more beneficial parts of the solution space results in a strong reduction of solution time and a slight reduction of solution quality. In case of a continuous practical application, the proposed combination of approaches will continuously extend the empirical basis and thus improve the performance of the MM approach by tendency.

References

1. Gössinger, R., Pishchulov, G., Dobos, I.: Order picking with heterogeneous technologies: an integrated article-to-device assignment and manpower allocation problem. In: Kliewer, N., Ehmke, J.F., Borndörfer, R. (eds.) OR Proceedings 2017, pp. 403–410. Springer, Cham (2018)
2. Gu, J., Goetschalckx, M., McGinnis, L.F.: Solving the forward-reserve allocation problem in warehouse order picking systems. J. Oper. Res. Soc. **61**, 1013–1021 (2010)
3. Walter, R., Boysen, N., Scholl, A.: The discrete forward-reserve problem – allocating space, selecting products, and area sizing in forward order picking. Eur. J. Oper. Res. **229**, 585–594 (2013)
4. Davis, D.J., Mabert, V.A.: Order dispatching and labor assignment in cellular manufacturing systems. Decis. Sci. **31**, 745–771 (2000)
5. Egilmez, G., Erenay, B., Süer, G.A.: Stochastic skill-based manpower allocation in a cellular manufacturing system. J. Manuf. Syst. **33**, 578–588 (2014)
6. Akagi, F., Osaki, H., Kikuchi, S.: A method for assembly line balancing with more than one worker in each station. Int. J. Prod. Res. **21**, 755–770 (1983)
7. Araújo, F.F.B., Costa, A.M., Miralles, C.: Two extensions for the ALWABP: parallel stations and collaborative approach. Int. J. Prod. Econ. **140**, 483–495 (2012)
8. Çengil, M.F., Albey, E., Yilmaz, G.: A hierarchical approach for assembly line balancing and competent worker assignment problem. In: Grubbström, R.W., Hinterhuber, H.H., Lundquist, J.E. (eds.) Proceedings of 20th International Working Seminar on Production Economics, vol. 2, pp. 129–140. Linköping (2018)
9. Barton, R.R., Meckesheimer, M.: Metamodel-based simulation optimization. In: Henderson, S.G., Nelson, B.L. (eds.) Handbooks in OR and MS, vol. 13, pp. 535–574. Amsterdam, North-Holland (2006)
10. Kleijnen, J.P.C.: Regression and Kriging metamodels with their experimental designs in simulation: a review. Eur. J. Oper. Res. **256**, 1–16 (2017)
11. McCormick, G.P.: Computability of global solutions to factorable nonconvex programs: part I - convex underestimating problems. Math. Prog. **10**, 147–175 (1976)

12. Khuri, A.I., Mukhopadhyay, S.: Response surface methodology. WIREs Comp. Stat. **2**, 128–149 (2010)
13. Nguyen, A.T., Reiter, S., Rigo, P.: A review on simulation-based optimization methods applied to building performance analysis. Appl. Energy. **113**, 1043–1058 (2014)

Fleet Sizing and Empty Freight Car Allocation

Philipp Hungerländer and Sebastian Steininger

1 Introduction

Railway freight transportation is characterized by various types of costs. One of the major cost drivers is the company's rolling stock, i.e. the locomotives and freight cars. In terms of the freight cars, the costs can be described in a simplified way as the expenses connected to the fleet size (holding costs per car per day), plus the costs of the actual movement of the fleet in the network. Therefore, fleet sizing problems are usually concerned with the determination of the minimum required number of vehicles needed to satisfy the demand of a transportation network. Once a consignment is shipped to its destination, the empty freight cars have to be reallocated. This reallocation process produces costs depending on traveled distance which can be minimized with the help of empty freight car distribution optimization.

In fact, the described types of cost drivers directly affect each other. The larger the fleet, the more freight cars are located all over the network and therefore, the less empty vehicle kilometers are typically needed, since some of the freight cars are close to the next customer. In turn, the smaller the fleet size, the longer the distance to the next customer and therefore, the distribution costs of empty cars increase. The aim of our proposed optimization model is to find the best trade-off between fleet size and empty vehicle kilometers in order to reduce the overall costs of a specific part of the distribution network of Rail Cargo Austria.

P. Hungerländer · S. Steininger (✉)
Alpen-Adria-Universität Klagenfurt, Klagenfurt, Austria
e-mail: philipp.hungerlaender@aau.at; sebastian.steininger@railcargo.com; sebastian.steininger@gmx.net

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_36

2 Literature Review

Several studies regarding empty vehicle allocation in the railway industry have been conducted. White and Bomberault [1], Jordan and Turnquist [2], Kikuchi [3] , Joborn et al. [4] and Mendiratta and Turnquist [5] introduced different network flow models. A taxonomy and literature review regarding empty freight car flow models were presented by Dejax and Crainic [6].

Fleet sizing problems have been studied in different fields of transport. Zuckerman and Tapiero [7] tried to determine the optimal fleet size for passenger transports, Jordan and Turnquist [8] developed a deterministic model for container fleet sizing, Beaujon and Turnquist [9] tried to find the optimal number and location of freight car pools and Sherali and Tuncbilek [10] compared a static and a dynamic model for fleet sizing for the automobile and railroad industries.

There already exist a few optimization models that jointly consider fleet sizing and empty freight car allocation. Bojovic [11] formulated a general system theory approach for minimizing overall costs. Sayarshad and Ghoseiri [12] developed an integer linear program (ILP) for fleet sizing and empty freight car allocation with deterministic demands which can solve small-size instances to optimality. Additionally, a simulated annealing heuristic was applied for solving real-world sized problems. Sayarshad and Tavakkoli-Moghaddam [13] proposed a model for optimizing fleet size and empty freight car allocation under uncertain demands. Both in [12] and [13] travel times were assumed to be deterministic and backordering of demand was included.

Since loading and unloading times may have a major impact on the overall time a freight car is idle, and therefore not available for further disposition, service times are taken into account in the model presented in the paper at hand. Thus, considerations of service times represent one of the main differences to the models proposed in [12] and [13]. Contrary to [13] we assume the demand of loaded freight cars to be deterministic. Additionally, demand has to be fulfilled completely in every period, i.e. the possibility of backlogging demand is not considered in our model. Moreover, a minimum and maximum restriction regarding the number of distributed empty freight cars per shipment is applied, due to economic reasons as well as conditions of certain route sections. We present an ILP formulation that is able to obtain optimal solutions for our large-scale real-world instances that stem from a specific part of Rail Cargo Austria's distribution network.

3 Mathematical Formulation

The network is divided into origin nodes $i \in I$ and destination nodes $j \in J$ of empty freight cars. Note that the origin nodes $i \in I$ are at the same time destinations of loaded freight cars. The total number of time periods in which demand could occur is denoted by $\mathcal{T} = \{1, \ldots, T\}$. In order to balance out the predefined demand for loaded freight cars $D_{j,i,t}$, shipped from j to i in t, a certain amount of empty

Table 1 Input parameters

Input parameters	Description
$d_{i,j}$	Distance in kilometers from origin i to destination j
cv	Costs per empty vehicle kilometer
cf	Holding costs per freight car per day
mV_j	Maximum capacity of freight cars at destination node j
mY_i	Maximum capacity of freight cars at origin node i
$\widetilde{W}_{i,j}$	Max. number of empty freight cars sent from i to j per distribution
$W_{i,j}$	Min. number of empty freight cars sent from i to j per distribution
$D_{j,i,t}$	Demand shipped from j to i in t
$\alpha_{j,i}$	Travel time from j to i
$\beta_{i,j}$	Travel time from i to j
δ_i	Service time for unloading in i
τ_j	Service time for loading in j

Table 2 Decision variables

Decision variables	Description
$X_{i,j,t}$	Number of empty freight cars sent from i to j in t
$S_{i,j,t}$	Number of empty freight cars sent from i arriving at j in t
$B_{j,t}$	Sum of new loaded freight cars at j in t
$P_{j,i,t}$	Number of loaded freight cars sent from j arriving at i in t
$L_{i,t}$	Sum of new empty freight cars at i in t
$Y_{i,t}$	Inventory of empty freight cars at i in t
$YS_{i,t}$	Inventory of loaded freight cars in service (unloading) at i in t
$V_{j,t}$	Inventory of loaded freight cars at j in t
$VS_{j,t}$	Inventory of empty freight cars in service (loading) at j in t
F	Fleet size

freight cars $X_{i,j,t}$ has to be distributed from i to j in t. Since freight cars could be distributed at the end of time horizon $\mathcal{T}$ and therefore arrive or be in service in $t > T$, we define another time horizon $\mathcal{T}_{DS} = \{1, \ldots, T_{DS}\}$ with $T_{DS} \geq \max\{T+\alpha_{j,i}+\delta_i\ ,\ T+\beta_{i,j}+\tau_j\}$. The time horizon for inventory is denoted as $\mathcal{T}_I = \{0, \ldots, T_I\}$ with $T_I \geq T$. In turn, the fleet size F can be defined as sum of all freight cars in inventory in $t = 0$. The total sets of input parameters as well as integer decision variables are summarized in Tables 1 and 2, respectively.

Now, we are able to formulate our fleet sizing and empty freight car allocation problem as a network flow problem on a time-space network, which can be modeled as the following integer linear program (ILP):

$$\min \sum_{t\in\mathcal{T}} \sum_{j\in J} \sum_{i\in I} d_{i,j} \cdot cv \cdot X_{i,j,t} + cf \cdot T \cdot F \tag{1}$$

$$\text{subject to: } F = \sum_{j\in J} VS_{j,0} + \sum_{j\in J} V_{j,0} + \sum_{i\in I} YS_{i,0} + \sum_{i\in I} Y_{i,0}, \tag{2}$$

$$S_{i,j,(t+\beta_{i,j})} = X_{i,j,t}, \qquad \forall\, i \in I, j \in J, t \in \mathcal{T}, \tag{3}$$

$$B_{j,(t+\tau_j)} = \sum_{i \in I} S_{i,j,t}, \qquad \forall\, j \in J, t \in \mathcal{T}, \tag{4}$$

$$VS_{j,t} = VS_{j,t-1} + B_{j,(t+\tau_j)} - B_{j,t}, \qquad \forall\, j \in J, t \in \mathcal{T}, \tag{5}$$

$$V_{j,t} = V_{j,t-1} + B_{j,t} - \sum_{i \in I} D_{j,i,t}, \qquad \forall\, j \in J, t \in \mathcal{T}, \tag{6}$$

$$V_{j,t} + VS_{j,t} \leq mV_j, \qquad \forall\, j \in J, t \in \mathcal{T}_I, \tag{7}$$

$$P_{j,i,(t+\alpha_{j,i})} = D_{j,i,t}, \qquad \forall\, j \in J, i \in I, t \in \mathcal{T}, \tag{8}$$

$$L_{i,(t+\delta_i)} = \sum_{j \in J} P_{j,i,t}, \qquad \forall\, i \in I, t \in \mathcal{T}, \tag{9}$$

$$YS_{i,t} = YS_{i,t-1} + L_{i,(t+\delta_i)} - L_{i,t}, \qquad \forall\, i \in I, t \in \mathcal{T}, \tag{10}$$

$$Y_{i,t} = Y_{i,t-1} + L_{i,t} - \sum_{j \in J} X_{i,j,t}, \qquad \forall\, i \in I, t \in \mathcal{T}, \tag{11}$$

$$Y_{i,t} + YS_{i,t} \leq mY_i, \qquad \forall\, i \in I, t \in \mathcal{T}_I, \tag{12}$$

$$W_{i,j} \leq X_{i,j,t} \leq \widetilde{W}_{i,j}, \qquad \forall\, i \in I, j \in J, t \in \mathcal{T}, \tag{13}$$

$$X_{i,j,t}, F \in \mathbb{Z}_{\geq 0}, \qquad \forall\, i \in I, j \in J, t \in \mathcal{T}, \tag{14}$$

$$S_{i,j,t}, P_{t,j,i}, L_{i,t}, l_{i,t}, b_{t,j}, B_{t,j} \in \mathbb{Z}_{\geq 0}, \qquad \forall\, i \in I, j \in J, t \in \mathcal{T}_{DS}, \tag{15}$$

$$Y_{i,t}, YS_{i,t}, V_{t,j}, VS_{t,j} \in \mathbb{Z}_{\geq 0}, \qquad \forall\, i \in I, t \in \mathcal{T}_I, \tag{16}$$

The objective function (1) is the minimization of overall costs. Accordingly, the optimal trade off between costs for empty vehicle kilometers per freight car sent from i to j and costs for freight car fleet per day is determined. Equation (2) defines the fleet size as sum of starting inventories in period $t = 0$. Constraints (3) state that empty freight cars dispatched from i to j in t will arrive at destination j after travel time $\beta_{i,j}$. Equation (4) ensure that the sum of all empty freight cars sent from i to j arriving in t are ready for redistribution in j after service time τ_j for loading. The inventory for empty freight cars currently in service (loading) $VS_{t,j}$, as well as the inventory for the loaded freight cars $V_{t,j}$ at j in t are defined by (5) and (6), respectively. Additionally, the inequalities in (7) restrict the overall inventory by the maximum capacity in node j. Similarly to Eq. (3) for empty freight cars, Constraints (8) state that loaded freight cars dispatched from j to i in t (demand), will arrive at i after the travel time $\alpha_{j,i}$. Equation (9) ensure that the sum of loaded freight cars sent from j to i arriving in t are ready for redistribution in i after the service time δ_i for unloading. Constraints (10)–(12) model the inventory change and the capacity restrictions at the origin node i in t and the inequalities in (13) define the limitations on the number of distributed empty freight cars from i to j in t. Constraints (14)–(16) ensure that all decision variables are non-negative integers.

4 Computational Experiments

In this section we test the proposed ILP using a set of input parameters motivated by a real-world monthly demand scenario of Rail Cargo Austria with 31 periods and a demand of 3994 cars to distribute. The total network consists of 14 origin nodes (A–N) and 14 destination nodes (1–14). Vehicle holding costs cf are set to 32 € per day and the costs for an empty vehicle kilometer are $cv = 1$ €. This setting is used for computing a baseline scenario. Furthermore, the impact of modifications from this baseline scenario concerning cost structure, travel time between certain nodes and a potential production stop are tested. Additionally, impacts of changes of time horizon and the number of nodes in the network on computing time are examined. The time limit was set to 1 h. All experiments were performed using IBM ILOG CPLEX Optimization Studio Version 12.8.0 on a Windows 7 64-bit machine equipped with an Intel Core i5-6300U CPU (2x2400MHz) and 4 GB RAM.

Table 3 summarizes the results of the experiments regarding different modifications in the network. With the help of the first three scenarios, concerning changes of cost structure, it can be shown how the optimal fleet size and empty kilometers needed affect each other, as rising costs for empty vehicle kilometers cause an increase of the fleet size and a reduction of the number of empty kilometers needed. In terms of modifications of network structure and demand it can be seen that small changes of parts of the network can affect the solution on the whole transportation network. In Table 4, we examine the relation between the problem size and computing time needed to solve the ILP. While smaller instances can be solved in a few seconds, computing time for bigger instances increases strongly.

Table 3 Effects of network modification

Fleet size	Empty kilometers	Objective value (€)	Modification
876	631,941	1,500,933	cf: 32 €, cv: 1 € (baseline)
912	603,278	1,281,806	cf: 24 €, cv: 1 €
974	566,460	2,099,128	cf: 32 €, cv: 2 €
1184	477,368	1,835,632	cf: 24 €, cv: 2 €
913	613,415	1,519,111	Double travel time on A-2 and F-8
877	618,667	1,488,651	Production stop at A in $t = 9 \ldots 15$

Table 4 Problem size and computing times

Origin nodes	Destination nodes	Periods	Demand	CPU time	Objective value
14	14	15	1904	00:04	958,543 €
14	14	31	3994	04:00	1,500,933 €
14	14	40	5127	16:48	1,798,127 €
14	14	80	5756	TO	3,094,560 € (0.13%)
20	20	31	4574	34:24	1,798,270 €
30	30	31	5926	TO	2,395,112 € (0.14%)

The time out (TO) is set to 1 h. In case of TO the upper bound and gap (%) are stated

However, with the proposed ILP it is still possible to solve the considered real-world problem with 28 nodes and 31 time periods in reasonable time.

5 Conclusion

In this paper we presented an integer linear programming formulation for fleet sizing and empty vehicle allocation in railway freight transportation developed for Rail Cargo Austria. With the help of the proposed model, the best trade-off between holding costs for freight cars and costs for empty vehicle kilometers can be determined. Changes in cost structure, travel times, demand, fleet size as well as empty vehicle kilometers were analyzed. Furthermore, the effects of the network size and time horizon on the computing time were tested. It was shown that bigger instances derived from a real-world scenario can be solved to optimality in an adequate time.

References

1. White, W.W., Bomberault, A.M.: A network algorithm for empty freight car allocation. IBM Syst. J. **8**(2), 147–169 (1969)
2. Jordan, W.C., Turnquist, M.A.: A stochastic, dynamic network model for railroad car distribution. Transp. Sci. **17**(2), 123–145 (1983)
3. Kikuchi, S.: Empty freight car dispatching model under freight car pool concept. Transp. Res. B Methodol. **19**(3), 169–185 (1985)
4. Joborn, M., Crainic, T.G., Gendreau, M., Holmberg, K., Lundgren, J.T.: Economies of scale in empty freight car distribution in scheduled railways. Transp. Sci. **38**(2), 121–134 (2004)
5. Mendiratta, V.B., Turnquist, M.A.: Model for management of empty freight cars. Trans. Res. Rec. **838**, 50–55 (1982)
6. Dejax, P.J., Crainic, T.G.: Survey papera review of empty flows and fleet management models in freight transportation. Transp. Sci. **21**(4), 227–248 (1987)
7. Zuckerman, D., Tapiero, C.S.: Random vehicle dispatching with options and optimal fleet size. Transp. Res. B Methodol. **14**(4), 361–368 (1980)
8. Turnquist, M.A., Jordan, W.C.: Fleet sizing under production cycles and uncertain travel times. Transp. Sci. **20**(4), 227–236 (1986)
9. Beaujon, G.J., Turnquist, M.A.: A model for fleet sizing and vehicle allocation. Transp. Sci. **25**(1), 19–45 (1991)
10. Sherali, H.D., Tuncbilek, C.H.: Static and dynamic time-space strategic models and algorithms for multilevel rail-car fleet management. Manag. Sci. **43**(2), 235–250 (1997)
11. Bojović, N.J.: A general system theory approach to rail freight car fleet sizing. Eur. J. Oper. Res. **136**(1), 136–172 (2002)
12. Sayarshad, H.R., Ghoseiri, K.: A simulated annealing approach for the multi-periodic rail-car fleet sizing problem. Comput. Oper. Res. **36**(6), 1789–1799 (2009)
13. Sayarshad, H.R., Tavakkoli-Moghaddam, R.: Solving a multi periodic stochastic model of the rail–car fleet sizing by two-stage optimization formulation. Appl. Math. Model. **34**(5), 1164–1174 (2010)

An Iterated Tabu Search for the Vehicle Routing Problem with Multiple Compartments and Last-in-First-Out Unloading

Felix Tamke

1 Introduction

The vehicle routing problem (VRP) and its various extensions belong to the best known problems in the operations research community. In recent years, researchers tend to approach problems which include more and more practical properties. We study a variant of the VRP where vehicles may have multiple compartments. The usage of multiple compartments is economically reasonable if different types of products should be transported on one truck. This is common in industries such as fuel delivery [8], grocery distribution [3], and waste collection [2]. Latter introduce the multi-compartment VRP with flexible compartment sizes (MCVRP-FCS) for a glass waste collection problem. A special feature of this problem is that compartment sizes are flexible but may just be varied in discrete steps. Additionally to the characteristics of the MCVRP-FCS in [2], we consider in this paper that all products must be loaded in compliance with a last-in-first-out (LIFO) unloading policy. Moreover, trucks need not return to the depot and the cost function is based on tariffs.

These features are part of a problem recently introduced in [7] and occur as a special case of a distribution problem at a large German food retailer. The company has to supply a vast number of hypermarkets with various goods from a central warehouse on a daily basis. Due to the different transport temperatures, these goods can be divided into four product categories: deep-frozen, chilled, slightly chilled, and unchilled. The delivery process is performed in cooperation with several freight forwarders which operate non-overlapping forwarder areas with multiple markets. The routing is planned by staff at the warehouse, whereas the forwarders provide the

F. Tamke (✉)
TU Dresden, Dresden, Germany
e-mail: felix.tamke@tu-dresden.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_37

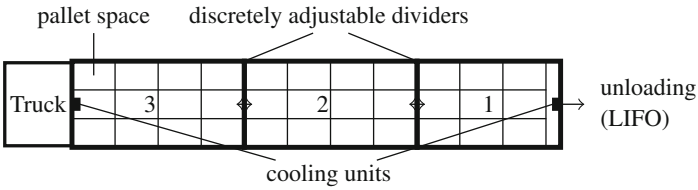


Fig. 1 Structure of a vehicle with three compartments and a capacity of 33 pallet spaces

necessary number of vehicles and execute the delivery process. As shown in Fig. 1, trailers can be partitioned into up to three compartments to transport more than one product category at once.

However, solely the outer compartments can be used for cooled goods, as cooling units are located at the front and rear ends. Additionally, compartment sizes are flexible, but may only vary in discrete steps. The step size is three as dividers are positioned orthogonal to the direction of traffic and three pallets can be loaded side by side. Besides, two further characteristics result from the cooperation with forwarders. On the one hand, vehicles do not have to return to the warehouse. On the other hand, forwarders are paid according to previously negotiated tariffs. These are dependent on both market and product category. This means that the tariffs of a forwarder differ between markets and also between the product categories of a market. Markets far away from the depot are usually more expensive than nearby markets and product categories that require refrigeration are more costly than uncooled products.

However, the tariffs are not sufficient to determine the costs of a route. The total costs of a route consist of three parts. The first part is a so-called base price. This corresponds to the most expensive market-product category combination delivered to any market on this route. Secondly, there are fixed stop costs for each additional stop after the first market. And finally, the detour of a route is multiplied with a fixed rate to consider the length of a route and to avoid combining far-flung markets. The detour is defined as the difference between the actual length of a route and the direct distance to the market furthest from the depot visited on that route. However, a detour below a certain threshold is included in the tariffs and is therefore not taken into account. In summary, this cost function differs a lot from the common distance or travel time minimization used in VRPs. We refer to [7] for an in depth explanation of the considered problem and the tariff-based cost function as well as a mixed-integer programming model.

In the following, we will give a detailed description of our solution approach in Sect. 2 and provide results of our computational experiments in Sect. 3. Finally, some further research directions are outlined in the last section.

2 The Iterated Tabu Search Approach

We propose an iterated tabu search (ITS) heuristic to tackle the problem described above. Our ITS approach is partly based on the ITS of [6] for the MCVRP, which is in turn inspired by the ITS algorithm of [1] for various VRPs. ITS is a variant of the well known iterated local search (ILS) framework introduced in [4] and was firstly applied in a routing context to the traveling salesman problem by [5]. As with ILS, ITS alternates between local search improvement phases for intensification and perturbation phases for diversification. However, in contrast to common ILS approaches, the local search part consists of a tabu search (TS) algorithm instead of local descent heuristics.

The general procedure of our ITS is presented in Algorithm 1. Firstly, we create an initial solution s_0 with single-customer routes for each market and improve s_0 by applying the TS for the first time. Subsequently, the generated solution s is set as first incumbent solution. Afterwards, the perturbation mechanism comprising two different components and the TS alternate with one another for n^{ITS} iterations. At the end of each iteration i, the best solution found s^* is updated, if the objective function value $f(s)$ is smaller than the known minimal costs $f(s^*)$. If not, we replace the current solution s with the incumbent solution s^* with probability $1-\left(i/n^{\text{ITS}}\right)^2$. This should support diversification at the beginning and intensification at the end as explained in [6]. The performance of an ITS algorithm is highly dependent on the TS as well as the perturbation mechanism. Therefore, our approach for these two parts is described in detail below.

TS itself is a meta-heuristic which employs local search methods to exploit a solution space. We explore the solution space by moving from the current solution to the best feasible, non-tabu solution in its neighborhood.

The neighborhood in our procedure *TabuSearch* is defined by shifting the complete or partial demand of a customer from one route to another. Thus, as shown in Fig. 2, a move consists of a source route (T1), a sink route (T2), a customer,

Algorithm 1 Iterated tabu search

1: $s_0 \leftarrow$ Initial solution with single-customer routes for each market
2: $s \leftarrow \mathit{TabuSearch}(s_0,\ n^{\text{TS}},\ \tau)$
3: $s^* \leftarrow s$
4: **for** $i = 1$ to n^{ITS} **do**
5: $\quad s' \leftarrow \mathit{SplitPerturbation}(s,\ \alpha)$
6: $\quad s'' \leftarrow \mathit{SwapPerturbation}(s',\ \beta)$
7: $\quad s \leftarrow \mathit{TabuSearch}(s'',\ n^{\text{TS}},\ \tau)$
8: $\quad$ **if** $f(s) < f(s^*)$ **then**
9: $\quad\quad s^* \leftarrow s$
10: $\quad$ **else if** random number $\in [0, 1] < \left(i/n^{\text{ITS}}\right)^2$ **then**
11: $\quad\quad s \leftarrow s^*$
12: $\quad$ **end if**
13: **end for**
14: Return best found solution s^*

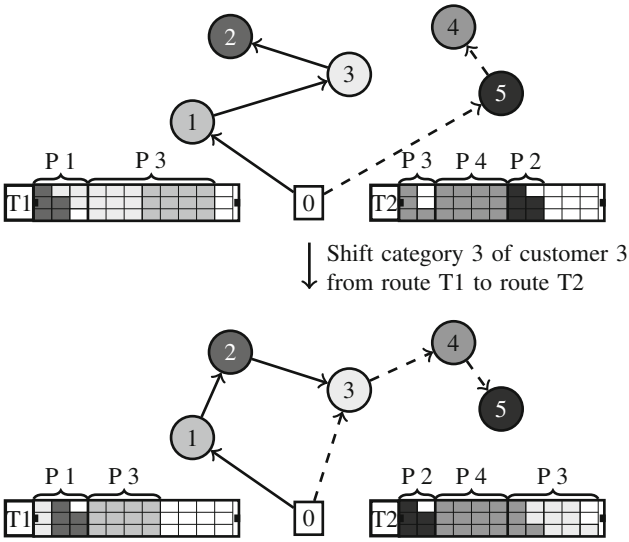


Fig. 2 Move operator of *TabuSearch*

and a non-empty set of products associated with this customer. Furthermore, we determine for each move the best feasible sequence of customers for both new routes. This is viable and necessary due to different reasons. First of all, the demands of each product category are relatively high in comparison to the vehicle capacity. This limits the number of customers per route as splitting the demand of a product category is not allowed. In addition, the best sequence may change although the set of visited customers remains the same as shown in Fig. 2. On route T1, the shortest sequence $1 \rightarrow 2 \rightarrow 3$ is not feasible before the move due to the LIFO-unloading policy and the structure of the trailer. However, after shifting category 3 of customer 3 to route T2, the best sequence is possible. The check for a feasible loading of a given sequence is implemented as depth-first search, where each product category of each customer corresponds to a node. The search terminates if a feasible loading is found or the customer sequence is proven infeasible. In addition to the LIFO constraints, a feasible loading has to respect the given capacity for discrete compartment sizes and ensure valid assignments of product categories to compartments.

We prohibit shifting recently relocated complete customer-product combinations for τ iterations to diversify the search. For example, if product categories two and three of customer one are selected to be relocated in the current iteration, it is tabu to move them together for the next τ iterations. However, they may be shifted separately. Tabu length τ is randomly selected from a given interval at the beginning of each TS. In addition, we apply the common aspiration criterion that a tabu move is permitted in case it would lead to a new best solution s of the current TS run.

Finally, *TabuSearch* stops if $f(s)$ has not improved for $n^{\mathrm{TS}} = \sqrt{\left(n^{\mathrm{ITS}} - i\right)\sqrt{n}}$, with n as number of customers, consecutive iterations and s is returned.

TabuSearch gradually reduces the number of vehicles used. This may result in suboptimal solutions with no more feasible non-tabu moves available as the demand-capacity ratio is relatively high. Since we only apply feasible moves, the TS would easily get stuck in a local optimum if the perturbation is not strong enough. Hence, our perturbation mechanism consists of two steps. Firstly, we employ a procedure called *SplitPerturbation* that increases the number of routes and generates solution s'. During this process, we destroy a percentage α of all divisible routes and create new single-customer routes for every product category of each customer. A route is considered as divisible if it is not a single-customer route with only one product. As we just split routes, it is not unlikely that the TS would return to the previous local optimum. To prevent this, we use the second perturbation step *SwapPerturbation*. This procedure performs β random swaps of two demands of product categories assigned to different routes. Like the tabu length, the perturbation strength β is selected randomly from a given interval. The resulting solution s'' is passed to *TabuSearch* again.

3 Computational Experiments

In the following, we assess our ITS approach by comparing its solutions with solutions obtained by solving a variant of the mixed-integer program (MIP) introduced in [7] with a commercial solver. We employ real data provided by the food retailer to generate a set of test instances. As stated in Sect. 1, all markets are allocated to one specific forwarder area. Hence, the general problem is divided into smaller subproblems for each area. We create 20 test instances with varying demands for six different areas A to F with five, seven, nine, ten, eleven, and twelve markets. Thus, 120 instances are considered in total. The demand of each product category of each market is randomly generated by using historical data.

Note that with a capacity of 33 pallet spaces per trailer, the average demand-capacity ratio per customer is larger than one for all areas. This results in a greater number of vehicles used in a solution than customers per instance exist, which is unusual for most VRPs. Additional data provided by the retailer include tariffs, real road network distances, cost rate for additional stops, costs per distance unit of a detour as well as the detour threshold.

The ITS as well as the MIP have been implemented in C++ using Microsoft Visual Studio 2017. Gurobi 8.0 has been applied as commercial solver for the MIP. All tests are performed on a Windows Server 2012 R2 with four Intel Xeon E5-4627 v2 3.3 GHz processors with 8 cores each and 768 GB RAM. However, only one core for the ITS, four cores for Gurobi as well as a small fraction of the available main memory has been used for testing. Preliminary tests of different parameter values for the ITS have found best performances for tabu length $\tau \in [5, 15]$, split factor

Table 1 Mean and best relative deviation in percent and runtime in seconds for 10^2, 10^3, 10^4 and 10^5 ITS iterations as well as relative deviation for Gurobi

	10^2			10^3			10^4			10^5			
Area	Mean	Best	t[s]	Mean	Best	t[s]	Mean	Best	t[s]	Mean	Best	t[s]	Gurobi 8.0
A	1.02	**0.00**	0.22	0.01	**0.00**	1.34	**0.00**	**0.00**	18.25	**0.00**	**0.00**	288.40	**0.00**
B	0.88	0.13	0.32	0.14	0.05	3.35	0.02	**0.00**	62.27	**0.00**	**0.00**	1175.51	0.21
C	2.23	1.01	0.45	0.85	0.04	5.29	0.16	0.05	111.70	0.03	**0.00**	2457.63	1.37
D	1.15	0.48	0.50	0.32	0.19	5.56	0.14	0.02	120.05	0.02	**0.00**	2760.40	0.83
E	1.81	0.45	0.56	0.30	0.14	6.51	0.13	0.07	139.16	0.07	**0.00**	3084.07	2.09
F	1.48	0.70	0.70	0.58	0.19	7.58	0.18	0.06	166.16	0.06	**0.00**	3892.20	1.79
Avg.	1.43	0.46	0.46	0.37	0.10	4.94	0.11	0.03	102.93	0.03	**0.00**	2276.37	1.05

$\alpha = 25\%$, and number of swap moves $\beta \in [0, 10]$. Each instance is solved 20 times with different seeds by our ITS approach to cope with the randomness of the solution method. The MIP is solved once for each instance with a time limit of 2 h. We provide the initial solution with single-customer routes as start solution since the solver is not able to find a feasible solution for some instances within the given time frame.

Computational results for 10^2, 10^3, 10^4 and 10^5 ITS iterations as well as for Gurobi are presented in Table 1. We measure the quality of a solution by its relative deviation (in percent) from the best known solution (BKS) of an instance. Table 1 shows for all ITS iterations and each forwarder area (averaging all instances) the mean relative deviation (Mean), the relative deviation of the best solution found during 20 runs (Best), and the average run time in seconds (t[s]). Since the MIP is solved just once, only the average relative deviation over all instances is presented for each area. Deviations with value zero are marked bold.

As expected, the solution quality improves and the run time rises with an increasing number of iterations. However, even for 10^2 iterations, which is solved in under 1 s on average, the mean relative deviation of all areas is just 1.43% and 2.23% in the worst case (C). If we apply 10^5 iterations, our ITS approach is always able to find the BKS and the mean relative objective value deviates by just averaging 0.03% from the BKS. Though, average run times are quiet high. In comparison with the solutions provided by Gurobi, the ITS is only worse if we use 10^2 iterations. In all other cases we can find better solutions on average with our ITS approach. This difference is particularly large for the biggest areas E and F. In addition, the majority of the BKS of the smallest area A are proven to be optimal by Gurobi. This shows that our algorithm generates high quality solutions for the given set of instances.

4 Conclusion and Further Research

We introduced a variant of the MCVRP-FCS with a LIFO unloading policy, no return of vehicles to the depot, and a real-world cost function based on tariffs. This distribution problem occurs at a large German food retailer. An iterated tabu

search with two perturbation mechanisms was presented to solve this challenging VRP. Computational tests with instances, which were generated by using real data provided by the retailer, showed that our algorithm can provide high quality solutions and performs better than applying a commercial solver as Gurobi to a mixed integer program.

Further studies may include improvements of the iterated tabu search, such as heuristic approaches for determining feasible customer sequences, considering only a limited neighborhood, and adaptive parameter management. Additionally, the algorithm may be extended to solve the entire problem described in [7]. Finally, the benefits of deliveries across forwarder areas may be explored by omitting the assumption on non-overlapping forwarder areas.

References

1. Cordeau, J.-F., Maischberger, M.: A parallel iterated tabu search heuristic for vehicle routing problems. Comput. Oper. Res. **39**(9), 2033–2050 (2012)
2. Henke, T., Speranza, M.G., Wäscher, G.: The multi-compartment vehicle routing problem with flexible compartment sizes. Eur. J. Oper. Res. **246**(3), 730–743 (2015)
3. Hübner, A., Ostermeier, M.: A multi-compartment vehicle routing problem with loading and unloading costs. Transp. Sci. **53**(1), 282–300 (2018)
4. Lourenço, H.R., Martin, O.C., Stützle, T.: Iterated local search. In: Glover, F., Kochenberger, G.A. (eds.) Handbook of Metaheuristics, chapter 11, pp. 320–353. Springer, Boston (2003)
5. Misevičius, A.: Using iterated tabu search for the traveling salesman problem. Inf. Technol. Control **32**(3), 29–40 (2004)
6. Silvestrin, P.V., Ritt, M.: An iterated tabu search for the multi-compartment vehicle routing problem. Comput. Oper. Res. **81**, 192–202 (2017)
7. Tamke, F.: A mixed-integer programming model for a vehicle routing problem in the food retailing industry. Dresdner Beiträge zur Betriebswirtschaftslehre **181**, 18 (2018)
8. Vidović, M., Popović, D., Ratković, B.: Mixed integer and heuristics model for the inventory routing problem in fuel delivery. Int. J. Prod. Econ. **147**, 593–604 (2014)

Part IX
Metaheuristics

An Adaptive Large Neighborhood Search Heuristic for Jointly Solving Storage Location Assignment and Picker Routing Problem

Necati Aras and Berk Görgülü

1 Introduction

A major cost component in operating a warehouse results from the movement of order pickers since order picking activities have a share of about 50–60% in the time spent in a warehouse [3], and traveling constitutes almost 50% of an order picker's time [9]. Hence, any attempt to decrease the total distance traveled by the pickers contributes to the reduction of the total cost. Two main factors affecting the total distance travelled by the pickers are the storage locations of the items and the routes followed by the pickers in collecting items.

Picker routing problem (PRP) is concerned with the minimization of the total traveling distance of the order pickers while collecting the items that make up the orders. It is an operational level problem, which assumes that the locations of the items stored in the warehouse have already been determined. The well-known heuristics proposed for walking along the aisles of a warehouse with only front and back cross aisles are traversal [5], return, mid-point, largest gap [6], and combined-walk heuristics [7, 8]. The assignment of items to locations in the warehouse is another optimization problem, which is referred to as the storage location assignment problem (SAP) in the literature. As opposed to the PRP, the SAP is a tactical level problem faced by the warehouse management since the storage locations assigned to the items cannot be changed frequently.

There are basically three types of storage assignment policies: random, dedicated and shared. In the *random* policy, incoming lots of items from suppliers are assigned to an empty location with equal probability. In the *dedicated* policy each item is

N. Aras (✉) · B. Görgülü
Boğaziçi University, Istanbul, Turkey
e-mail: arasn@boun.edu.tr; http://www.boun.edu.tr/~aras

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_38

stored in the same address. When the item order rates vary over time, the dedicated policy may become inconvenient depending on the level of fluctuation in the order rates. This gives rise to the *shared* storage policies where items are divided into classes, the warehouse is partitioned into zones, and each class of items is then stored in a given zone. Another consideration is that items ordered together should be stored close to each other so as to reduce the distance traversed for order picking. The so-called correlated storage assignment problem (CSAP), which is shown to be an NP-hard problem in [4], takes into account the complementarity of items. Due to the complexity of exactly solving the CSAP for large-sized instances encountered in real-world situations, the authors develop a heuristic procedure for its solution. They compute first the correlation among the items and then generate clusters of items based on their correlation similarity. There exist several papers examining the CSAP such as [1, 2, 10].

To the best of our knowledge, PRP and SAP have not been solved before in an integrated fashion even though many papers have been published that focus on one of these problems, and consider the other one in an indirect fashion. We formulate a novel mixed-integer linear programming (MILP) model for the joint storage assignment and picker routing problem (JSAPRP). It can provide an optimal solution for small-sized instances and a feasible solution for medium-sized instances, but fails to give any feasible solution for large-sized instances. Therefore, we develop a heuristic solution method based on adaptive large neighborhood search (ALNS) that performs an efficient search in the space of the storage assignments and computes the objective value by solving a Traveling Salesman Problem (TSP) to find a route for the picker for each order.

The rest of the paper is organized as follows. Section 2 presents a mathematical programming model for the JSAPRP. Section 3 describes the ALNS heuristic. Computational results are presented in Sect. 4. Conclusions and some remarks are given in Sect. 5.

2 A Mathematical Model for Joint Storage Assignment and Picker Routing Problem

In this section, we develop an MILP model for the JSAPRP. Let $i, j \in I$ be the index of storage locations, 0 the depot location (entrance, input/output point), I_0 the set of storage locations and the depot location, $o \in O$ the index of orders, and $k \in K$ the index of items. Parameter $a_{ok} = 1$ if item k exists in order o, $a_{ok} = 0$ otherwise. Parameter h_{ok} denotes the volume of item k existing in order o, and d_{ij} represents the traveling distance between location i and location j. Parameter q is the volume capacity of a pallet. Binary decision variable $X_{oij} = 1$ if location j is visited after location i while picking items existing in order o, $Y_{ki} = 1$ if item k is located at location i. Continuous variable F_{oij} is the volume of the pallet during

traveling from location i to location j while picking items of order o.

$$\min \sum_{o \in O} \sum_{i \in I_0} \sum_{j \in I_0, j \neq i} d_{ij} X_{oij} \tag{1}$$

subject to

$$\sum_{k \in K} Y_{ki} = 1 \qquad i \in I \tag{2}$$

$$\sum_{i \in I} Y_{ki} = 1 \qquad k \in K \tag{3}$$

$$\sum_{j \in I_0} X_{oij} = \sum_{k} a_{ok} Y_{ki} \qquad o \in O, i \in I \tag{4}$$

$$\sum_{j \in I_0} X_{oji} = \sum_{k} a_{ok} Y_{ki} \qquad o \in O, i \in I \tag{5}$$

$$\sum_{i \in I} X_{o0i} = \sum_{i \in I} X_{oi0} \qquad o \in O \tag{6}$$

$$X_{oij} + \sum_{k} a_{ok} Y_{ki} - \sum_{k} a_{ok} Y_{kj} \leq 1 \qquad o \in O, i, j \in I, i \neq j \tag{7}$$

$$X_{oij} + \sum_{k} a_{ok} Y_{kj} - \sum_{k} a_{ok} Y_{ki} \leq 1 \qquad o \in O, i, j \in I, i \neq j \tag{8}$$

$$\sum_{j \in I_0, j \neq i} F_{oij} = \sum_{j \in I_0, j \neq i} F_{oji} + \sum_{k \in K} h_{ok} Y_{ki} \qquad o \in O, i \in I \tag{9}$$

$$F_{oij} \leq q X_{oij} \qquad o \in O, i, j \in I_0, i \neq j \tag{10}$$

$$X_{oij} + X_{oji} \leq 1 \qquad o \in O, i, j \in I_0, i \neq j \tag{11}$$

$$X_{oij} \in \{0, 1\}, F_{oij} \geq 0 \qquad o \in O, i, j \in I_0, i \neq j \tag{12}$$

$$Y_{ki} \in \{0, 1\} \qquad i \in \mathcal{I}, k \in K \tag{13}$$

The objective function (1) represents the total traveling distance required for picking all the orders. Constraints (2) ensure that only one item is assigned to a given location, while constraints (3) guarantee that every item is assigned to exactly one location. Constraints (4) make sure that if a location is visited while picking an order (the right-hand side is equal to one), then the picker has to travel from this location to another one including the depot. On the other hand, if the right-hand side is zero, then no travel is possible to another location during order picking. Constraints (5) imply that if a location is visited, then the picker has to travel to this location from another one including the depot. Constraints (6) act as a valid inequality and force the same number of pallets (i.e., one) must leave and

return to the depot for each order. Constraints (7) and (8) imply that the picker can travel from a location to another one if and only if both locations have to be visited consecutively while picking the items of an order. Constraints (9) are the flow balance constraints written for each location. Constraints (10) ensure that if a picker travels from location i to location j, then the volume of the pallet must not exceed its capacity. Constraints (11) eliminate subtours of length one.

3 Nested Adaptive Large Neighborhood Search Heuristic

It is clear that the heuristic solution procedure to be developed should incorporate decisions related to both subproblems SAP and PRP. To this end, we devise a nested heuristic that performs at the outer level a search over the storage assignment decisions using an adaptive large neighborhood search (ALNS) heuristic. It determines at the inner level the routes of the pickers for every order by solving a TSP by means of tabu search. The various move operators applied in ALNS help to explore a large neighborhood of the current solution. Note that the nested structure allows us to consider the routes that the pickers traverse during the determination of the best storage locations for the items.

After an initial solution is generated by means of a greedy construction algorithm, it is improved by applying seven different move operators, some of which use the following three measures: (1) order rate OR_k of item k. It denotes the ratio of the number of orders including a particular item to the total number of orders during a predefined period of time; (2) joint order rate of two items. JOR_{kl} measures how frequently two particular items k and l co-exist in orders; (3) vote of an item which is defined as $V_k = OR_k + \sum_{l \in L \setminus \{k\}} JOR_{kl}$, where L is the set of items having a certain property. The move operators are defined as follows. Random 2-Swap: Two items are selected randomly (i.e., with equal probability) from the current solutions, and their locations are exchanged. Random 3-Swap: Three items are selected randomly from the current solution, and their locations are changed in such a way that all of them are assigned to new locations. Notice that there exist two possible ways of doing this, and one possibility is chosen randomly. HiJOR 2-Swap: An item k is selected randomly, and then item l with the highest joint order rate JOR_{kl} with item k is determined. The locations of these two items are exchanged. LoJOR 2-Swap: An item k is selected randomly, and then item l with the lowest joint order rate JOR_{kl} with item k is determined. The locations of these two items are exchanged. Closest OR 2-Swap: An item k is selected randomly, and then item l with the closest order rate OR_l to OR_k of item k is determined. The locations of these two items are exchanged.

At each iteration, candidate solutions (item-to-location assignment configurations) are created first using the move operators. To compute the objective value corresponding to a candidate assignment configuration, a TSP has to be solved for each order to determine the route that gives the sequence of the items to be

picked in that order. This implies that as many TSPs as the number of orders should be solved for each candidate item-to-location assignment configuration. In ALNS, the moves are selected with probabilities, and moves that lead to better solution are selected with a higher probability. To implement this idea, we define a weight w_m for each move type m. Moreover, let n_m denote the number of times move m is selected, and s_m represents the historical success of move m. Initially, the weight of all move operators is set to one, i.e., $w_m = 1$. The initial values of parameters n_m and s_m are set to zero. At each iteration, a move operator is selected randomly based on its selection probability that is proportional to its weight. The parameters are updated based on the quality of the new solution generated in the neighborhood of the current solution by the selected move operator.

4 Computational Results

By setting four different values to $|K| = \{10, 20, 30, 40\}$, three different values to $|O| = \{10, 20, 30\}$, and creating three versions for each combination as a result of varying the items existing in each order, we generate a total of 36 small instances. Results are displayed in Table 1 for only one version of each combination to save space. Here, LB stands for a lower bound on the optimal objective value (OV) of the instance and UB represents the OV of the best feasible solution (an upper bound on the optimal OV) both attained by Cplex 12.8. Since we solve each instance five times using ALNS, we report both the best OV obtained in these runs as well as an average value. The CPU time of the ALNS heuristic indicates the computation time for each run. PD in the last column of the table represents the percent deviation of the best OV given by the ALNS heuristic from the best OV provided by Cplex, where negative values imply that ALNS yields a better solution than Cplex. The results reveal that the quality of the solutions found by ALNS is quite promising. Eight of the smallest nine instances with $|K| = 10$ and $|O| = \{10, 20, 30\}$ can be solved to optimality within a considerable shorter amount of time than the Cplex solver. As the size of the instances gets larger, ALNS is able to obtain better solutions than Cplex in less CPU time. On the average, the best (average) solution of the ALNS heuristic outperforms Cplex by 41.0% (40.4%) over 36 instances, which also indicates the robustness of the ALNS heuristic.

We also generate new instances much larger than the earlier ones with $|O| = 300$ orders and $|K| = \{100, 200, 300, 400, 500\}$ items. Since it is not possible to compare our heuristic with Cplex, we want to find out how much improvement can be obtained by applying the ALNS heuristic within 3 h when we start with an initial solution based on a greedy construction heuristic. For five instances we compute the improvements of 8.3%, 8.9%, 9.6%, 9.4%, and 9.1% with an average of 9%.

Table 1 Performance comparison of ALNS and Cplex on small instances

	Cplex				ALNS heuristic			
Instance	LB	UB	Gap (%)	CPU time (s)	Best OV	Average OV	CPU time (s)	PD (%)
K10-O10	211.50	211.50	0.00	415	211.50	213.21	4.98	0.00
K10-O20	380.80	380.80	0.00	1322	380.80	384.55	7.50	0.00
K10-O30	589.20	589.20	0.00	7141	589.20	590.80	11.24	0.00
K20-O10	329.28	433.40	24.02	10,800	410.80	425.11	347	−5.21
K20-O20	507.92	896.95	43.37	10,800	706.45	720.01	295	−21.24
K20-O30	832.76	1747.95	52.36	10,800	1255.00	1264.83	657	−28.2
K30-O10	414.05	1031.40	59.86	10,800	640.85	650.59	1357	−37.87
K30-O20	709.00	1951.35	63.67	10,800	1239.55	1257.80	1608	−36.48
K30-O30	890.39	2765.00	67.8	10,800	1957.45	1989.08	1800	−29.21
K40-O10	540.99	1409.75	61.63	10,800	948.60	955.65	1800	−32.71
K40-O20	711.38	3686.45	80.7	10,800	1647.85	1667.20	1800	−55.3
K40-O30	1218.58	6556.85	81.42	10,800	2825.90	2828.73	1800	−56.9

5 Conclusion

In this paper, we address an important problem faced in warehouses, namely the joint storage location assignment and picker routing problem. We propose two solution methods for the solution of this problem: a mixed-integer linear programming model that can handle small instances only, and a heuristic method based adaptive large neighborhood search. The heuristic's performance is shown to be very promising compared with the state-of-the-art solver Cplex on small instances. On large instances, the heuristic can improve the solution of a greedy construction heuristic by approximately 9%.

References

1. Bindi, F., Manzini, R., Pareschi, A., Regattieri, A.: Similarity-based storage allocation rules in an order picking system: an application to the food service industry. Int. J. Log. Res. Appl. **12**(4), 233–247 (2009)
2. Chiang, D.M.H, Lin, C., Chen, M.: Data mining based storage assignment heuristics for travel distance reduction. Expert Syst. Appl. **31**(1), 81–90 (2014)
3. Frazelle, E.: World-Class Warehousing and Material Handling. McGraw Hill, NewYork (2002)
4. Frazelle, E., Sharp, G.: Correlated assignment strategy can improve order-picking operation. Ind. Eng. **4**, 33–37 (1989)
5. Goetschalckx, M., Ratliff, H.: Order picking in an aisle. IIE Trans. **20**(1), 53–62 (1998)
6. Hall, R.: Distance approximations for routing manual pickers in a warehouse. IIE Trans. **24**(4), 76–87 (1993)
7. Ratliff, H., Rosenthal, A.: Order picking in a rectangular warehouse: a solveable case of the traveling salesman problem. Oper. Res. **31**, 507–521 (1983)
8. Roodbergen, K.: Layout and routing methods for warehouses. Ph.D. thesis, Rotterdam School of Management, Erasmus University (2001)
9. Tompkins, J., White, J., Bozer, Y., Frazelle, E., Tanchoco, J.: Facilities Planning. Wiley, Hoboken (2003)
10. Zhang, Y.: Correlated storage assignment strategy to reduce travel distance in order picking. IFAC-PapersOnLine **49**(2), 30–35 (2016)

Predicting the Vibroacoustic Quality of Steering Gears

Paul Alexandru Bucur, Klaus Frick, and Philipp Hungerländer

1 Introduction

The propagation of quality specifications across the supply chain is challenging in domains where quality can be subjective, such as in automotive acoustics. In the current work, we direct our attention to a situation encountered in the daily operations of one of the world's leading steering system suppliers, ThyssenKrupp Presta AG, where requirements imposed on the vibroacoustic quality of the steering gear need to be passed down to its subcomponents. Furthermore, only one subcomponent, the ball nut assembly (BNA), is subject to an own vibroacoustical quality test equivalent to the one of the steering gear.

In the current production setting, the vibrational signals of the BNA are transformed to the frequency domain and analyzed as order spectra. For each order spectrum curve, acoustic domain experts determine a set of order intervals and corresponding quality thresholds. Between the orders corresponding to the left and right boundaries of each interval, the maximum value of the order spectrum curve must lie below the threshold, otherwise the BNA is marked as being qualitatively not ok and thus destined to be scrapped or reworked (see Fig. 2 for an intuitive visualization).

This approach operates under the assumption that single orders in the steering gear order spectra are influenced by single orders in the BNA order spectra. If any steering gear order violates its threshold in its own test, then, in the case of a BNA fault, the corresponding BNA order must be itself subject to a stricter threshold.

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

K. Frick
Institute of Computational Engineering, NTB Interstate University of Technology Buchs, Buchs, Switzerland

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_39

In the present work, we propose a more flexible methodology, which uses as input only the BNA order spectra and the information whether the steering gear failed its test due to the BNA, classifying thus the BNA in *OK* and *NOK*. An advantage of this relaxation is that the boundaries and their respective thresholds can be optimized and learned from the data, considering also possible non-linear effects in the vibroacoustical transfer path between the BNA and steering gear. Furthermore, the restriction on a limited number of spectral intervals, each equipped with a constant intensity threshold, can be considered as a regularization strategy against overfitting. As such, deriving the classifier amounts to optimizing an appropriate accuracy measure over the set of piecewise constant functions, with the design variables of the optimization problem being the interval boundaries, the respective threshold values and the number of intervals.

Similar optimization problems have been extensively considered in the context of piecewise constant regression, where a given time series is approximated by piecewise constant functions. This problem is often referred to as *multiple change point problem* [1, 2]. An essential shortcoming of these approaches for our application is the assumption that the cost function $E(f)$ can be expressed as the sum of interval costs, i.e. $E(f) = \sum_{j=0}^{k} E\left(f_{|I_j}\right)$, where f is a piecewise constant function with k change points and intervals I_j. For this situation, efficient optimization algorithms such as dynamic programming [3] or binary segmentation [4] have been proposed. In our case, these approaches are not applicable, due to the fact that if the spectral threshold of any interval is violated, the cost is already maximal.

Our main contribution is twofold: first, we solve a real-world multiple change point problem by developing an algorithm which hybridizes a custom genetic algorithm and the Nelder-Mead downhill simplex method. This results in a better performance w.r.t. the Cohen Kappa metric than the currently employed method and in the reduction of the solution complexity, due to the usage of less change points. Second, we speed up computations by guiding the genetic mutations using a weight function gained via deep canonical correlation analysis between the ball nut assemblies and the corresponding steering gear.

The remainder of the paper is organized as follows: in Sect. 2 we describe the novel problem formulation, followed by the description of the hybrid algorithm and its use of the canonical correlation weights in Sect. 3. In Sect. 4 we discuss the computational experiments and their results, concluding the work in Sect. 5 by mentioning possible directions for further research on the topic.

2 Problem Formulation

We shall henceforth assume N BNA order spectra as input, sampled at K common, equidistant order indices. The spectral intensities are thus given as a matrix $X \in \mathbb{R}^{N \times K}$.

Mathematically speaking, the current quality examination method described in Sect. 1 amounts to binary classification of spectral signals. A signal is classified *OK* (positive class) if it is majorized by a predefined piecewise constant function and *NOK* else. Let $s_1, \ldots, s_K$ be the grid of spectral orders at which the signals are sampled. We call f a *piecewise constant function with k change-points* if there exist indices $1 = i_0 < i_1 < \ldots < i_k < i_{k+1} = K$ and values $f_0, \ldots, f_k$ such that $f(s) = f_j$ if $s \in (s_{i_j}, s_{i_{j+1}}]$. We further denote by $\mathcal{P}_k$ the set of all piecewise constant functions with k change-points.

In this work we aim to find an optimized function $f \in \mathcal{P}_k$, where optimality is expressed by means of two fitness measures. Due to our highly imbalanced dataset, in which the *OK* class has a prevalence of $p = 0.965$, we choose *Cohen's Kappa* [5] as a classification accuracy metric for the BNA and thus as the first fitness function, accounting for the solution quality:

$$\kappa(f) := \frac{p_o - p_e}{1 - p_e}, \tag{1}$$

where p_o denotes the relative observed agreement and p_e represents the hypothetical probability of chance agreement between the binary labels and the quality predictions resulting from the application of f.

In practice, the quantity $\kappa(f)$ is estimated from the training data. We refer to this estimated value as $\kappa_{\text{train}}(f)$. We further take an interest in the business aspect of the problem, since a faulty BNA assembled into a steering gear is ten times as expensive as a BNA that is mistakenly destroyed. To this effect, we introduce $E(f) = 10\text{FPR}(f) + \text{FNR}(f)$, where $\text{FPR}(f)$ and $\text{FNR}(f)$ denote the false positive and false negative rates; this metric is used only to evaluate the business impact, but not for optimization purposes. The second objective function aims to reduce the total number of employed intervals.

With these definitions at hand, we formulate the multiple change point optimization problem:

$$\kappa^*(k) = \kappa_{\text{train}}(f^*(k)) = \max_{f \in \mathcal{P}_k} \kappa_{\text{train}}(f), \quad k \in \mathbb{N}, \tag{2}$$

where the best-performing number of change points is denoted by k^* and the corresponding piecewise constant function by $f^* \in \mathcal{P}_{k^*}$.

3 Proposed Algorithm

In order to solve problem (2) we maximize κ_{train} for a fixed k w.r.t. the threshold values and the change-point positions. We pursue an alternate direction approach [6], iteratively maximizing w.r.t. the thresholds while keeping the change-points fixed and vice versa. The maximization w.r.t. to the change-points is carried out by

a genetic algorithm and the maximization w.r.t. to the threshold values by means of the Nelder-Mead downhill simplex method, with the complete algorithm sketched in Algorithm 1.

input : N order spectrum curves and the corresponding binary labels
number of change points k
maximal number of generations B
output: Optimized piecewise constant threshold function $f^* \in \mathcal{P}_k$

Initialize population $\pi[0]$ of 100 randomly generated functions in $\mathcal{P}_k$;
for *i from* 1 *to* B **do**
  Update current population
  for f *in* π **do**
    Update threshold values for f such that $\kappa_{\text{train}}(f)$ is maximized while keeping the change points constant.
  end
  Breed next generation
  $\pi[i] = \text{evolve}(\pi[i-1])$
end

Algorithm 1: Genetic algorithm for finding an optimized thresholding function

This procedure is repeated for $k = 1, 2, \ldots$, with the k^* value manually chosen so as to optimize the trade-off between solution quality (*Cohen's Kappa*) and solution complexity (number of change points); the corresponding $f^* \in \mathcal{P}_{k^*}$ is the final proposed solution. In the following we describe the individual substeps in Algorithm 1 in more detail.

In the **update** step a given function $f \in \mathcal{P}_k$ is modified by adapting its threshold values $(f_0, \ldots, f_k)$ while keeping the change point positions constant, such that $\kappa_{\text{train}}(f)$ is maximized. Since we deal with a data-driven objective function, the mapping $f \mapsto \kappa_{\text{train}}(f)$ is not differentiable w.r.t to the threshold values. We thus employ the Nelder-Mead downhill simplex method, a gradient-free maximization technique. We stop the iteration when the increments in κ_{train} and f_i both are less than 10^{-4}.

The **evolution** step follows classical approaches in genetic optimization [7]. In each iteration we retain those 20% of the population individuals with the highest fitness value for reproduction and add 5% randomly generated elements in $\mathcal{P}_k$. Pairs of step functions are chosen from these individuals at random with replacement and crossed-over to generate offspring individuals, which are appended to the population until the original population size is reached. A proportion of 15% of the offspring individuals are mutated after generation.

The **crossover** method is rather straightforward: The k change points of the two parent functions are merged and an agglomerative cluster algorithm is run on the $2k$ positions. The resulting dendrogram is then cut such that k clusters result, with the centroids of these clusters chosen as the offspring individual's change points. The values between the new change points are obtained by averaging the function values of the parent functions over the new intervals.

In the **mutation** step the change point locations are randomly shifted with a probability of 0.15, under the constraint of preservation of order. The maximal allowed deviation is additionally constrained depending on the order. The results of previous work by means of deep canonical correlation analysis [8] showed that some parts of the BNA order spectrum are more influential on the acoustic property of the stearing gear than others. We use the correlation coefficients as weight functions in the mutation routine, introducing larger change-point variation in relevant parts of the spectrum.

4 Computational Experiments

The dataset is composed of a total of 13,257 (9279 training and 3978 testing) BNA, from which 454 (305/149) are *NOK*. The population size for the genetic algorithm is fixed at 100, with the maximal number of generations B fixed at 500. In Fig. 1 the κ-values κ_{train} and κ_{test} are plotted against the number of change-points. Together with the company and considering the quality-complexity tradeoff on the κ_{train} values, we identified 9 as the best-performing number of change points instead of the 14 currently employed, with the optimized piecewise constant function f^* depicted in Fig. 2. The usage of f^* clearly increases the classification accuracy w.r.t. the Cohen Kappa metric, from the $\kappa_{\text{test}}(f^\dagger) = 0$ value resulting from the usage of the currently employed piecewise constant function $f^\dagger$ to a value of $\kappa_{\text{test}}(f^*) = 0.283$. While $\kappa = 0$ indicates only coincidental agreement, values in the range $(0.2, 0.4]$ represent already fair agreement. At the same time, the business costs on the test data are reduced by 24.16% from an initial value of $E(f^\dagger) = 1490$ to $E(f^*) = 1130$.

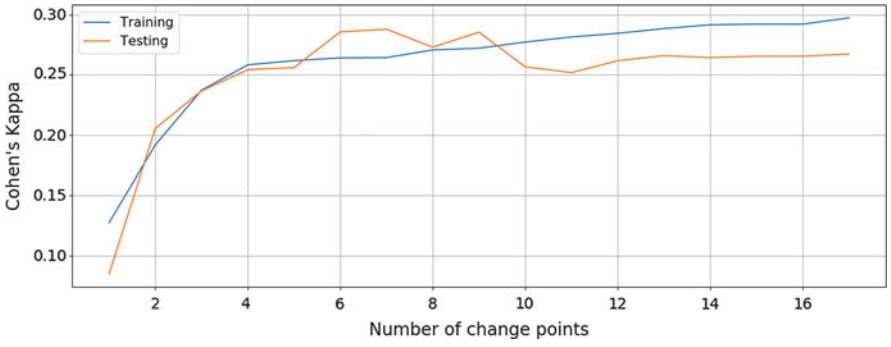


Fig. 1 Optimized Cohen's Kappa value in dependence of the number of change points k, computed for $f^*(k)$ on the training and test data (Color figure online)

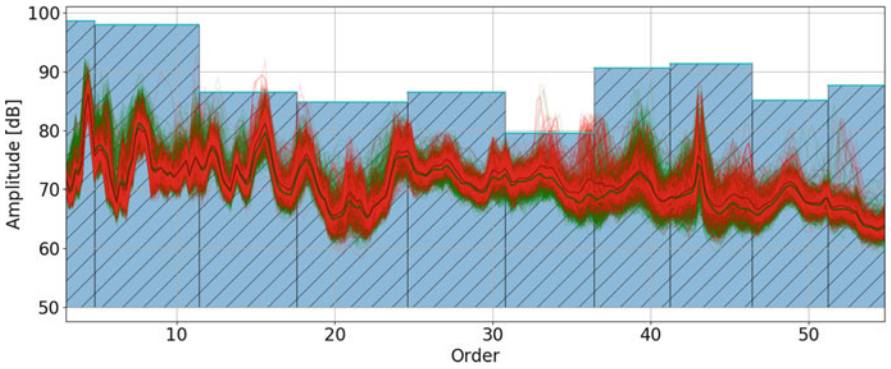


Fig. 2 Best piecewise constant function $f^* \in \mathcal{P}_9$ together with *OK* (green/dark) and *NOK* (red/light-coloured) BNA order spectra (Color figure online)

5 Conclusion

The cooperation with ThyssenKrupp Presta AG showed the potential of mathematical optimization techniques for real-world industrial problems. The combination of genetic algorithms and canonical correlation analysis allowed the optimization of the currently employed BNA quality examination method, increasing substantially its performance w.r.t. the Cohen Kappa metric and reducing at the same time its number of used change points and thus its ultimate complexity. The developed method is currently being evaluated for introduction into worldwide production systems, with the plausibility of the canonical weights already confirmed by acoustic domain experts. In an extended version of this work we aim to present more complex methods which are potentially able to further improve the results.

References

1. Niu, Y.S., Hao, N., Zhang, H.: Multiple change-point detection: a selective overview. Stat. Sci. **31**(4), 611–623 (2016)
2. Frick, K., Munk, A., Sieling, H.: Multiscale change point inference. J. R. Stat. Soc. Ser. B (Stat. Methodol.) **76**(3), 495–580 (2014)
3. Friedrich, F., Kempe, A., Liebscher, V., Winkler, G.: Complexity penalized M-estimation. J. Comput. Graph. Stat. **17**(1), 201–224 (2008)
4. Fryzlewicz, P.: Wild binary segmentation for multiple change-point detection. Ann. Stat. **42**(6), 2243–2281 (2014)
5. Ben-David, A.: Comparison of classification accuracy using Cohen's Weighted Kappa. Expert Syst. Appl. **34**(2), 825–832 (2008)
6. Niesenm U., Shah, D., Wornell, G.: Adaptive alternating minimization algorithms. In: 2007 IEEE International Symposium on Information Theory, pp. 1641–1645. IEEE, Piscataway (2007)

7. Mitchell, M.: An Introduction to Genetic Algorithms, vol. 1. MIT Press, Cambridge (1998)
8. Bucur, P.A., Frick, K., Hungerländer, P.: Correlation analysis between the vibroacoustic behavior of steering gear and ball nut assemblies in the automotive industry. In: EngOpt 2018 Proceedings of the 6th International Conference on Engineering Optimization, pp. 1253–1262. Springer, Cham (2019)

Part X
Optimization Under Uncertainty

Dynamic Policy Selection for a Stochastic-Dynamic Knapsack Problem

Jeannette Anna Lena Hermanns, Jan Brinkmann, and Dirk Christian Mattfeld

1 Introduction

As many optimization problems cannot be solved to optimality within polynomial time, heuristics are applied to identify reasonable solutions. In the field of stochastic-dynamic optimization, decisions are taken under uncertainty and over a sequence of decision points. An exogenous process reveals stochastic information stepwise. In every decision point, a policy yields a decision. Approximate dynamic programming (ADP, Powell [5]) provides a set of anticipatory policies autonomously adapting to stochastic information. As anticipation is challenging, the dynamic selection of policies promises improvement.

In a related article [1], we present an approach dynamically selecting parameters for a predefined policy by means of value function approximation (VFA). VFA carries out a priori simulations to learn suitable parameters. We show, that the parameter selection by VFA leads to significant improvement compared to manual selection and common benchmark policies. In this work, we generalize the approach by introducing the dynamic policy selection (DPS). The DPS draws on a set of predefined candidate policies. In every decision point, the DPS selects the most promising candidate policy.

The idea of the DPS has already been introduced in the field of deterministic-static optimization. According to Burke et al. [2], a hyper heuristic (HH) selects a candidate heuristic or combines heuristics (e.g., by means of genetic algorithms). The heuristics are selected or combined with respect to given instances. Then, instances are solved in one decision point without uncertainty. The DPS selects a candidate policy in every decision point of a stochastic-dynamic optimization

J. A. L. Hermanns (✉) · J. Brinkmann · D. C. Mattfeld
Decision Support Group, University of Braunschweig, Braunschweig, Germany
e-mail: j.hermanns@tu-braunschweig.de; j.brinkmann@tu-braunschweig.de; d.mattfeld@tu-braunschweig.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_40

problem with respect to stochastic distributions. Therefore, the contribution of this work is to adapt the idea of HHs for stochastic-dynamic optimization and to approve its benefits.

To evaluate our approach, we consider a stochastic-dynamic knapsack problem (SDKSP) where the decision maker has to accept or reject items in order to maximize the rewards. Computational studies point out that DPS leads to significant improvements compared to the individual use of the candidate policies. An analysis reveals that the solution structures reflect the stochastic distributions.

The remainder of this article is structured as follows: In Sect. 2, we define the SDKSP. The DPS and its candidate policies are introduced in Sect. 3. In Sect. 4, we explain the experiments and present the results of the computational study. We draw a conclusion in Sect. 5.

2 A Stochastic-Dynamic Knapsack Problem

The SDKSP (Papastavrou et al. [4]) is an extension of the static-deterministic 0-1-knapsack problem (KSP, Martello and Toth [3]). In the SDKSP, we have to accept or reject items that are revealed in a sequence. Every item comes with a reward and a weight. The rewards and weights are subject to a stochastic distribution. The sum of accepted items' weights must not exceed the knapsack's capacity. For every item, we immediately have to decide whether to accept or to reject. Future items are unknown until they appear. The goal is to maximize the total rewards of the accepted items.

We model the SDKSP as a Markov decision process (MDP, Puterman [6]). The MDP is a finite sequence of decision points. A decision point k occurs when an item of the sequence is revealed. The associated decision state $s_k \in S$ comprises information about the decision point in time $t_k \in [t_0, t_{\max}]$, the item's reward $r_k \in \mathbb{R}^+$, the items' weight $w_k \in \mathbb{R}^+$, and the knapsack's remaining capacity $c_k \in \mathbb{R}_0^+$: $s_k = (t_k, r_k, w_k, c_k)$. In every decision point, the decision $x \in X$ indicates whether the item is rejected or accepted. Accepting the kth item is feasible if the item's weight does not exceed the knapsack's remaining capacity: $w_k \leq c_k$. Therefore, $X = \{0, 1\}$. Else, $X = \{0\}$. In the subsequent post-decision state (PDS), the decision has already been implemented: $s_k^x = (t_k, c_k - x \cdot w_k)$. Then, new items are revealed until the last item $k_{\max}$ has been processed.

A policy $\pi \in \Pi$ yields a decision in every decision state: $\pi : S \rightarrow X$. The objective (1) is to identify an optimal policy $\pi^* \in \Pi$ maximizing the expected reward over all items conditioned on the initial decision state s_1:

$$\pi^* = \underset{\pi \in \Pi}{\arg\max}\, \mathbb{E}\left[\sum_{k=1}^{k_{\max}} \pi(s_k) \cdot r_k \middle| s_1\right]. \tag{1}$$

Since the SDKSP is $\mathcal{NP}$-hard, we cannot solve large instances to optimality in reasonable time. Thus, in Sect. 3, we draw on policies produced by ADP.

3 The Dynamic Policy Selection

The idea of the DPS is to combine candidate policies in order to gain benefits compared to the individual application of the candidate policies. Therefore, in every decision state, we select one of the following candidate policies that have different intentions.

Greedy The greedy policy accepts each item until the capacity is exhausted. The advantage is that the knapsack is always nearly full in the end. A disadvantage is that items of future decision points are ignored.

CBP The cost-benefit policy (CBP) is a rule of thumb accepting an item if the ratio of reward and weight exceeds a certain threshold. Let τ_k be the threshold in decision point k which is an adapted ratio over all items seen so far. Therefore, we recursively define τ_k according to Formula (2) such that we take the new items more into account than those appeared in the beginning.

$$\tau_k = \begin{cases} 1 & \text{, if } k = 1 \\ \frac{1}{2} \cdot \left(\frac{r_{k-1}}{w_{k-1}} + \tau_{k-1} \right) & \text{, otherwise} \end{cases} \tag{2}$$

Thus, an item is accepted if it is more profitable than the items seen before. The advantage is that this policy adapt to the distribution with minimal effort. The disadvantage is that future items are not explicitly considered and the threshold slowly adapt to changes of the item's distribution.

Anticipatory In addition to the current item, the anticipatory policy explicitly takes future items into account. Therefore, we draw on the value of the post-decision state s_k^x following s_k after decision x is made. The value $v_{s_k}^x \in \mathbb{R}_0^+$ is an approximation of the expected reward of accepted items in the future. As accepting an item reduces the remaining capacity, our decision influences the expected future rewards. We approximate the values $v_{s_k}^x$ by means of VFA [5]. To this end, we initialize each expected reward value with 0. Then, we carry out simulations of the MDP. In every decision state, the anticipatory policy makes the decision $x^\star \in X$ that maximizes the approximated outcome, i.e., the sum of current items reward and the expected future reward:

$$x^\star = \underset{x \in X}{\arg\max} \left\{ x \cdot r_k + v_{s_k}^x \right\} \tag{3}$$

After every simulation, we determine the average values for every PDS observed according to Formula (4). Here, $v_{s_k,\text{sim}}^x$ indicates the gained rewards after s_k^x in the

current simulation. The occurrences of s_k^x over all simulations are denoted by $\alpha_{s_k}^x$.

$$v_{s_k}^x := \frac{\alpha_{s_k}^x - 1}{\alpha_{s_k}^x} \cdot v_{s_k}^x + \frac{1}{\alpha_{s_k}^x} \cdot v_{s_k,\text{sim}}^x \tag{4}$$

The advantage of the anticipatory policy is that future items are explicitly considered. However, rare profitable items observed in certain stages of the time horizon may distort the values. This disadvantage results in unused capacity.

Policy Selection We have highlighted the advantages and disadvantages of the candidate policies in the previous paragraphs. Now, we define the DPS to take the advantages of all candidate policies $\pi \in \Pi'$. The DPS selects the policy leading to the maximum of expected future rewards. Unlike the anticipatory policy which operates on states and *decisions*, the DPS operates on states and *policies*. Here, the values $v_{s_k}^\pi \in \mathbb{R}_0^+$ approximate the expected future rewards if in state s_k, policy π is selected. Again, we initialize the expected reward values with 0 and carry out simulations by means of VFA. To control exploration and exploitation, we draw on the Boltzmann exploration [5]. To this end, in Formula (5), we assign every policy $\pi_i \in \Pi'$ a probability $\vartheta_i \in (0, 1)$ to be selected. The probabilities depend on the decision yield by the policy, the items' reward, the partially approximated values, and a temperature δ.

$$\vartheta_i = \frac{\exp\left(\left(\pi_i(s_k) \cdot r_k + v_{s_k}^{\pi_i}\right) \cdot \delta\right)}{\sum_{\pi \in \Pi'} \exp\left(\left(\pi(s_k) \cdot r_k + v_{s_k}^{\pi}\right) \cdot \delta\right)} \tag{5}$$

$$\delta = 1 + \frac{n_{s_k} \cdot 0.01}{\max_{\pi \in \Pi'}\{\left(\pi(s_k) \cdot r_k + v_{s_k}^{\pi}\right)\} - \min_{\pi \in \Pi'}\{\left(\pi(s_k) \cdot r_k + v_{s_k}^{\pi}\right)\}} \tag{6}$$

The temperature $\delta \in \mathbb{R}_0^+$ is defined in Formula (6). Generally speaking, exploration is enforced if δ is small. Let n_{s_k} indicate the number of occurrences of s_k over all simulations. Then, the procedure enforces exploration if s_k occurred rarely. Additionally, if the difference of approximated outcomes is large, inferior policies are allowed to be selected and, in this way, exploration is enforced. We update the average values $v_{s_k}^\pi$ according to Formula (7) after each simulation. In the current simulation, the rewards after the selection of π in s_k are indicated by $v_{s_k,\text{sim}}^\pi$. The total number of selection of π in s_k is denoted by $\alpha_{s_k}^\pi$.

$$v_{s_k}^\pi := \frac{\alpha_{s_k}^\pi - 1}{\alpha_{s_k}^\pi} \cdot v_{s_k}^\pi + \frac{1}{\alpha_{s_k}^\pi} \cdot v_{s_k,\text{sim}}^\pi \tag{7}$$

Eventually, when the approximation of values is completed, we can select the most promising policy $\pi^\star \in \Pi'$ as shown in Formula (8).

$$\pi^\star = \underset{\pi \in \Pi'}{\arg\max} \left\{\pi(s_k) \cdot r_k + v_{s_k}^\pi\right\} \tag{8}$$

4 Computational Studies

In the computational studies, we show evidence of the benefits of the DPS. We describe the experimental setup, present the results, and provide an analysis.

Setup In the computational studies, we assume that the stochastic information follow specific stationary distributions. Therefore we investigate four different distributions for the items' rewards and weights where the expectations of the normal distribution change over time. In Fig. 1, the abscissas show the time horizon. The ordinates show the expected rewards (dotted lines) and weights (dashed lines). Distributions *take late* and *take the middle* represent a low reward and a high weight in the beginning. Thus, mainly non-profitable items appear first. The ratio continuously inverts once for *take late* and twice for *take the middle*. Distribution *take early* mirrors *take late*, and *avoid the middle* mirrors *take the middle*. In every distribution, the number of possible items is 100. The times when the items appear are uniformly distributed. For the anticipatory policy and the DPS, we stop the approximation when no significant changes of values are observed. For the anticipatory policy, that is after 10^7 simulations on resampled training sequences of

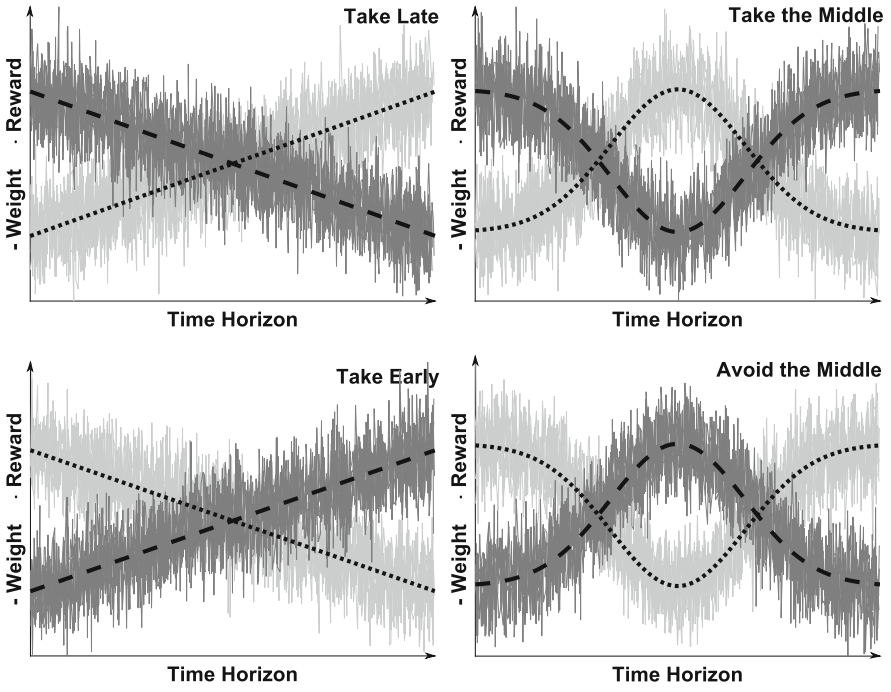


Fig. 1 The expected items' rewards and weights

Table 1 Overview of solution quality

Distribution	Greedy	CBP	Anticipatory	DPS
Take late	17.5	24.8	76.4	93.9
Take the middle	16.6	23.6	85.0	97.8
Take early	92.8	87.0	96.7	97.4
Avoid the middle	78.5	58.2	77.8	88.4

items, and 10^9 for the DPS. To accelerate the approximation process, we aggregate PDSs. First, we subdivide the time horizon into 120 periods and the capacity in intervals of size 1. Second, we summarize PDS which are in the same period and in the same interval. For every distribution, we resample 1000 test sequences to evaluate the candidate policies as well as the DPS.

Results We compare the average of results achieved with a common ex-post benchmark algorithm operating on perfect information. As introduced by Martello and Toth [3], the linear relaxation of the 0-1-KSP sorts all items by their reward-weight-ratio in descending order. Then, all items starting at the beginning of the order will be accepted until the capacity of the knapsack is exhausted. Table 1 depicts the percentages of achieved rewards for the candidate policies and the DPS compared to the benchmark. The results point out that the DPS leads to a higher solution quality compared to the candidate policies. The results are also demonstrating that the greedy policy is performing worse when the non-profitable items are coming up at the beginning (*take late* and *take the middle*). When the profitable items are coming in the beginning, the greedy policy is performing better. The performance of CBP is more stable but is not showing well results in any case. The anticipatory policy leads to the best results among candidate policies for *take late*, *take the middle*, and *take early*. It is slightly outperformed by the greedy policy in *avoid the middle*. Moreover, for every distribution, the results of the DPS are significant better compared to the individual usage of candidate policies.

Analysis Figure 2 illustrated the policies selected by the DPS for every distribution. The abscissas depict the periods. The ordinates show the percentages of policy selections and cases when the items are not acceptable due to exhausted capacity. For *take late*, we need anticipation to know that more profitable items appear late. For this reason, the anticipatory policy is predominant in the beginning. At the end, the CBP and the greedy policy are selected to accept the profitable items. In *take the middle*, non-profitable items mainly appear in the beginning and at the end of the time horizon. Profitable items appear in the middle. Here, CBP and the anticipatory policy are performing best just before the profitable items appear. Greedy is the best policy in the middle of the time horizon when the profitable items appear. At the end, the capacity is already exhausted. In *take early*, the profitable items appear in the beginning. Therefore, the greedy policy

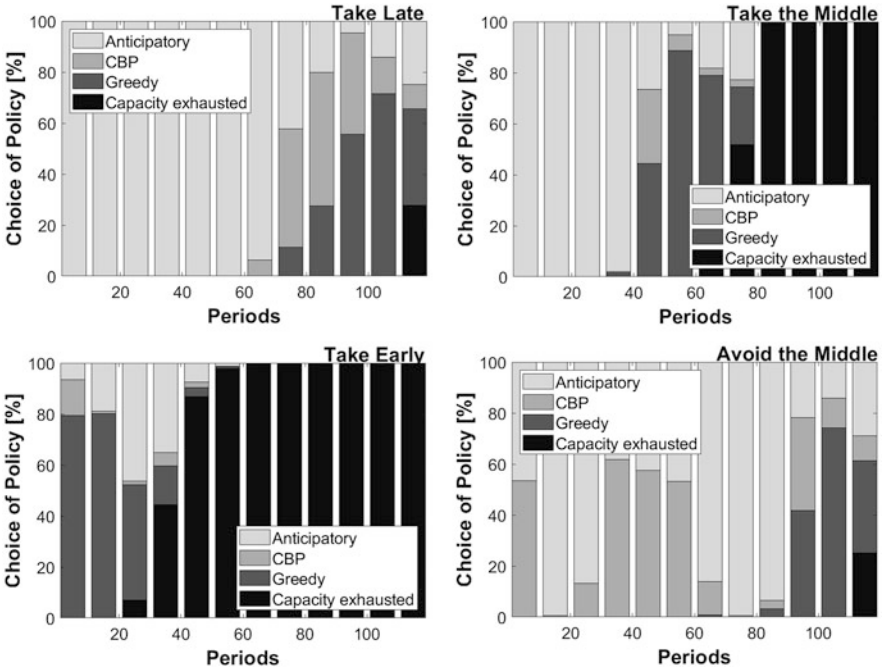


Fig. 2 Selected candidate policies by the DPS

is predominant at the beginning. Later, when the non-profitable items appear, the capacity is already exhausted. In *avoid the middle*, we face two patterns where profitable items appear: in the beginning and at the end. In most periods, the anticipatory policy and the CBP are selected to identify profitable and non-profitable items. In the end, i.e., in second buy pattern, the greedy policy is selected until the capacity is exhausted.

In *avoid the middle*, the two patterns of profitable items lead to challenges. Therefore, we provide a detailed view on the decisions made (to accept or to reject) by using the anticipatory policy or the DPS, respectively. At the top of Fig. 3, the decisions are displayed by using the anticipatory policy. At the bottom, the decision choices by the DPS are shown. The abscissas depict the periods. The ordinates show the percentages of the decision choices. The anticipatory policy mainly uses the first pattern to fill the knapsack. Consequentially, there is nearly no chance to take advantage from the second pattern. In contrast, the DPS accepts items in both patterns. This shows that the DPS can find meaningful patterns through the usage of dynamically selected candidate policies.

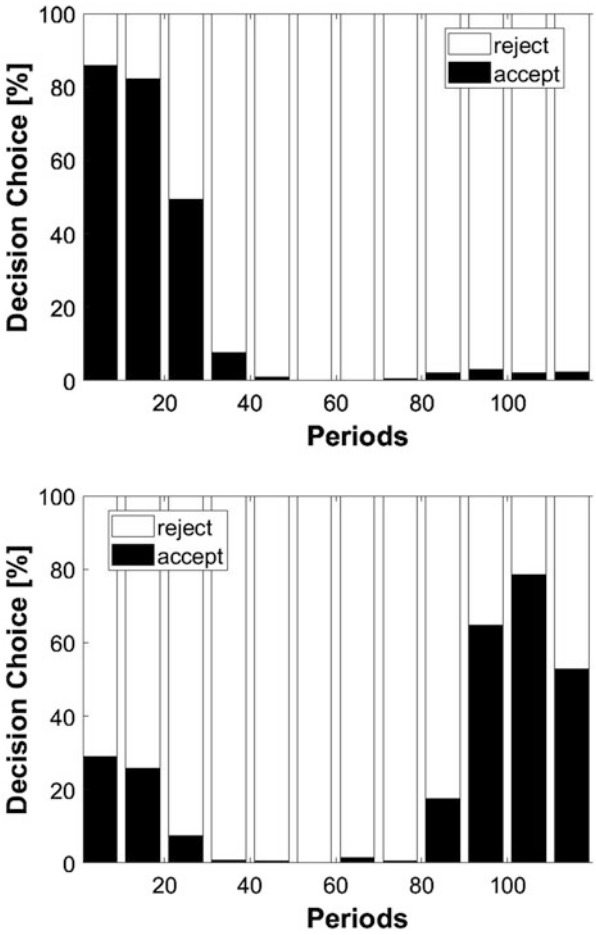


Fig. 3 Decisions for *avoid the middle* by the anticipatory policy and the DPS

5 Conclusion

In this paper, the advantage of combing different candidate policies to achieve better solution qualities for an $\mathcal{NP}$-hard stochastic-dynamic optimization problem is demonstrated. We exemplary draw on the SDKSP. The SDKSP is formulate as an MDP. In every decision point, an item appears which we either accept or reject. The item's reward and weight are subject to a stochastic distribution. The DPS dynamically selects a policy among a set of candidate policies. Computational studies point out that the DPS leads to a higher solution quality compared to the individual usage of candidate policies. Further, individual usage of policies misses to discover meaningful patterns in the distributions. The DPS, however, allows to reveal meaningful patterns and makes use of them. The proposed DPS approach

improves the solution quality for all of the presented problem instances. As the policies' contributions to the overall solution are state-dependent, the DPS takes advantage by dynamically selecting a policy in each state. Future research may focus on the adaption of the DPS for real-world applications, e.g., in production and logistics as well as in mobility.

Acknowledgements Funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—227198829 / GRK1931

References

1. Brinkmann, J., Ulmer, M.W., Mattfeld, D.C.: Dynamic lookahead policies for stochastic-dynamic inventory routing in bike sharing systems. Comput. Oper. Res. **106**, 260–279 (2018). https://doi.org/10.1016/j.cor.2018.06.004
2. Burke, E.K., Hyde, M., Kendall, G., Ochoa, G., Özcan, E., Woodward, J.R.: A classification of hyper-heuristic approaches. In: Handbook of Metaheuristics, vol. 146. Springer, Boston (2010). https://doi.org/10.1007/978-1-4419-1665-5
3. Martello, S., Toth, P.: Knapsack Problems: Algorithms and Computer Implementations. Wiley, Chichester (1990)
4. Papastavrou, J.D., Rajagopalan, S., Kleywegt, A.J.: The dynamic and stochastic knapsack problem with deadlines. Manag. Sci. **42**(12), 1706–1718 (1996). https://doi.org/10.1287/mnsc.42.12.1706
5. Powell, W.B.: Approximate Dynamic Programming: Solving the Curses of Dimensionality, 2nd edn. Wiley, Chichester (2011)
6. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming, 2nd edn. Wiley, Chichester (2014)

Part XI
OR in Engineering

Trajectory Optimization for Wire-Arc Additive Manufacturing

Armin Fügenschuh, Markus Bambach, and Johannes Buhl

1 Introduction

Due to the emerging new technology of additive manufacturing (AM), certain mechanical components that are currently manufactured by metal cutting operations could be manufactured without substantial material removal by AM processes. As AM process we consider the so-called wire-arc additive manufacturing (WAAM) process in this work. The wire is melted using high temperatures produced by an electrical arc and then transferred as droplets onto the workpiece. For simple geometries, the component is built layer-wise, form bottom to top. Parts that are more complex are decomposed and built sub-part by sub part, following their curvature [5]. In each layer, the welding torch deposits the molten wire droplet-by-droplet to the desired position. Path planning is crucial for process time and part quality. Deadheading is allowed if the shape in a certain layer cannot be drawn continuously. The most critical restrictions are due to the enormous heat input to the workpiece from the electrical arc. The heat is critical in the process since it leads to stress in the workpiece and thus is a potential source of deformations. For a single layer we thus describe the optimization problem of how to partition a given traverse into continuous segments that are manufactured without intersection and deadheading between two segments, such that deadheading is minimized in order to finish the WAAM process as fast as possible. As a further constraint of the optimization the accumulated heat is tracked and the local amount of heat should be below a given threshold. We give a formulation of these problems as a mixed-integer program and demonstrate the applicability of standard mixed-integer solvers for their solution on a test-bed of components.

A. Fügenschuh (✉) · M. Bambach · J. Buhl
Brandenburg University of Technology Cottbus-Senftenberg, Cottbus, Germany
e-mail: fuegenschuh@b-tu.de; bambach@b-tu.de; buhl@b-tu.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_41

2 Problem Description and Mathematical Observations

A prototypical workpiece to be manufactured is shown as a three-dimensional CAD plot in Fig. 1 (left). A single layer is shown on the right of this figure, where it is modeled using 17 individual segments, which are straight line segments or curved segments with one of the 12 nodes as end points.

In mathematical terms, this can be considered as an undirected graph. Ideally, one is able to traverse this graph by a consecutive path. Such a path through all segments would travers each single segment exactly once. Hence this manufacturing scheme would lead to a production of the segment in minimum time. As Euler conjectured in his famous problem of the Seven Bridges of Königsberg in 1736 (and Hierholzer proved in 1873), this is possible if and only if the degree of each node (that is, the number of incident segments in a node) is even with at most two exceptions allowed among all nodes (the so-called Euler property). In general, a workpiece is constructed with respect to a certain functionality, and thus one cannot expect that the Euler property holds. Hence deadheading or transiting from one node to another is needed which in mathematical terms relates our problem to the Chinese Postman Problem (CPP) [1]. Edmonds and Johnson gave the first integer programming formulation of the CPP, and several others were developed ever since for the classical version as well as for variants of it (see e.g., Sun et al. [6] for CPP with time-dependent travel times). The CPP has found its applications in mechanical engineering problems, for example, in torch path determination in flame cutting [4], which is close to our application (up to the fact that we do not cut away, but add material). In this sense, we describe here another variant of the CPP, with the temperature of the nodes as an additional feature. In modeling the path of the welding torch and the computation of the temperature our approach is related to Frank et al. [2]. Our model is further related to the design of cold-formed sheet metal workpieces as described in [3], where a flow formulation was used to place material in an optimal way under manufacturing constraints.

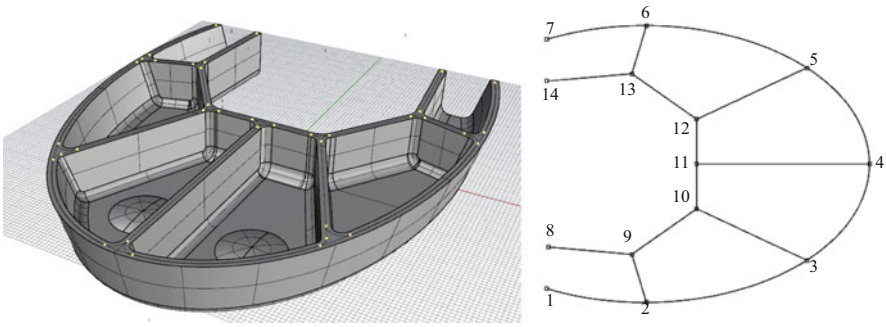


Fig. 1 The workpiece to be produced with WAAM in 3D (left), and a single layer of it with 12 nodes and 17 segments (right)

3 Model Formulations

A layer of the object to be printed is modeled as a graph (V, W), which consists of set of path segments W which intersect with other segments in some node i of the set V. A segment $\{i, j\} \in W$ can be printed in either direction, and thus we define a set $\overline{W} := \{(i, j), (j, i) : \{i, j\} \in W\}$. It is assumed that the welding torch manufactures a segment from start to end, and only at the end nodes it is decided how to continue the printing process: either by printing a consecutive segment that starts in this node, or by transiting to another position in V. In general, a transit can be carried out between any two nodes i and j, and thus the set of all transit arcs is given by $A := \{(i, j) \in V \times V : i \neq j\}$. All welding and transiting operations are carried out over a finite time horizon, which is denoted by $T := \{1, 2, \ldots\}$ and $T_0 := T \cup \{0\}$. An element in T describes a time step of a certain given constant length.

For each node $i \in V$ its cartesian coordinates $(\xi_i, \eta_i) \in \mathbb{R}^2$ are given (unit: meter). The transit distance between two points $(i, j) \in A$ is the euclidean distance: $d_{i,j} := \sqrt{(\xi_i - \xi_j)^2 + (\eta_i - \eta_j)^2}$. The segment length of $(i, j) \in \overline{W}$ is denoted by $\ell_{i,j}$. Since the segments are not limited to straight lines, we have that $\ell_{i,j} \geq d_{i,j}$. The speed of the welding torch is v^w when welding and v^d when deadheading or transiting (in meter per second). This results in a number of time steps that are required for the respective operation: $\Delta^w_{i,j} := \lceil \ell_{i,j}/v^w \rceil$ for $(i, j) \in \overline{W}$ and $\Delta^d_{i,j} := \lceil d_{i,j}/v^d \rceil$ for $(i, j) \in A$. The welding temperature of the arc wire is $\varphi^w \in \mathbb{R}_+$, the ambient temperature is w.l.o.g. assumed to be zero degree. We assume two (dimensionless) heat conduction coefficients of the material, denoted by $\kappa^w \in (0, 1)$ and $\kappa^e \in (0, 1)$. That is, if the material has a temperature of θ before welding in its neighborhood, it will be heated up to $\kappa^w \cdot \varphi^w + (1 - \kappa^w) \cdot \theta$ because of the heat induced by the welding torch. There is a heat loss of κ^e from one time step to the next, that is, if θ is the temperature of the material in some time step t, then in the next time step $t + 1$ it will be cooled down to $\kappa^e \theta$.

A node $i \in V$ is an odd node, if the number of incident arcs in W (with tail or head i) is an odd number. The number of odd nodes ω in a graph is always even. By a theorem of Euler from 1736, a closed path can be found through all segments, if at most two nodes are odd. In all other cases, transits are necessary, and the number of such transits is given by $\tau := \omega/2 - 1$.

We introduce variables $x_{i,j,t} \in \{0, 1\}$ for all $(i, j) \in \overline{W}$ and $t \in T$, with the meaning $x_{i,j,t} = 1$ if the segment (i, j) is manufactured in such ways, that the welding torch arrives in j at time step t. We further introduce variables $y_{i,j,t} \in \{0, 1\}$ for all $(i, j) \in A$ and $t \in T$, with $y_{i,j,t} = 1$ if the welding torch transits from i to j, and arrives in j at time step t. The temperature of each node $i \in V$ at each time step $t \in T_0$ is represented by $\theta_{i,t} \in \mathbb{R}_+$. The maximum temperature over all nodes is denoted by $\Theta \in \mathbb{R}_+$.

The problem of finding a production plan with a minimum maximal temperature over all nodes is then formulated as follows:

$$\min\ \Theta + \varepsilon \sum_{(i,j)\in A, t\in T} d_{i,j}\, y_{i,j,t}, \tag{1}$$

$$\text{s.t.} \sum_{(i,j)\in\overline{W}} x_{i,j,1} = 1, \tag{2}$$

$$\sum_{(i,j)\in\overline{W}} x_{i,j,|T|} = 1, \tag{3}$$

$$\sum_{(i,j)\in A, t\in T} y_{i,j,t} = \tau, \tag{4}$$

$$\sum_{t\in T} (x_{i,j,t} + x_{j,i,t}) = 1, \qquad \forall\, \{i,j\} \in W, \tag{5}$$

$$\sum_{(i,j)\in\overline{W}} x_{i,j,t} + \sum_{(i,j)\in A} y_{i,j,t} \le 1, \qquad \forall\, t \in T, \tag{6}$$

$$\sum_{i:(i,j)\in\overline{W}} x_{i,j,t} + \sum_{i:(i,j)\in A} y_{i,j,t} \le \sum_{k:(j,k)\in\overline{W}} x_{j,k,t+\Delta^w_{j,k}} + \sum_{k:(j,k)\in A} y_{j,k,t+\Delta^d_{j,k}}, \qquad \forall\, j \in V, t \in T, \tag{7}$$

$$\sum_{i:(i,j)\in A} y_{i,j,t} \le \sum_{k:(j,k)\in\overline{W}} x_{j,k,t+\Delta^w_{j,k}}, \qquad \forall\, j \in V, t \in T, \tag{8}$$

$$\kappa^w \varphi^w \sum_{(i,j)\in\overline{W}} x_{i,j,1} = \theta_{i,0}, \qquad \forall\, i \in V, \tag{9}$$

$$-\varphi^w \left(1 - \left(\sum_{i:(i,j)\in\overline{W}} x_{i,j,t} + \sum_{i:(i,j)\in A} y_{i,j,t}\right)\right) + \kappa^w \varphi^w + (1-\kappa^w)\kappa^e \theta_{j,t-1} \le \theta_{j,t}, \qquad \forall\, j \in V, t \in T, \tag{10}$$

$$\varphi^w \left(1 - \left(\sum_{i:(i,j)\in\overline{W}} x_{i,j,t} + \sum_{i:(i,j)\in A} y_{i,j,t}\right)\right) + \kappa^w\varphi^w + (1-\kappa^w)\kappa^e\theta_{j,t-1} \geq \theta_{j,t}, \qquad \forall\, j \in V, t \in T, \tag{11}$$

$$\kappa^e\theta_{j,t-1} - \varphi^w \left(\sum_{i:(i,j)\in\overline{W}} x_{i,j,t} + \sum_{i:(i,j)\in A} y_{i,j,t}\right) \leq \theta_{j,t}, \forall\, j \in V, t \in T, \tag{12}$$

$$\kappa^e\theta_{j,t-1} + \varphi^w \left(\sum_{i:(i,j)\in\overline{W}} x_{i,j,t} + \sum_{i:(i,j)\in A} y_{i,j,t}\right) \geq \theta_{j,t}, \forall\, j \in V, t \in T, \tag{13}$$

$$\theta_{i,t} \leq \Theta, \qquad \forall\, i \in V, t \in T_0. \tag{14}$$

The objective function (1) minimizes the maximal temperature in each node Θ. As a subordinate goal, unnecessary transits shall be avoided. For this, ε is set to 10^{-3}. The production plan starts with the manufacturing of some segment (2), and also ends in this way (3). Having a transit either at the beginning or at the end would be suboptimal and thus can be explicitly excluded by these constraints. The number of transits can be determined from the number of odd nodes (4). Each segment must be manufactured at a certain point in time, in either direction (5). In each time step, the welding torch is either active or transiting (6). The path of the welding torch is a consecutive sequence of production and transition steps (7). After a transiting step, the welding torch must be activated again, i.e., a transiting must not follow another transiting move (8). The initial temperature for node i is either ambient temperature or the welding temperature of the arc wire (9). The temperature in a node equals the temperature in the previous time step plus the energy gain from a welding torch heat exposure nearby (10), (11). In case there is no welding torch, the node temperature depends only on the temperature of the previous time step (12), (13). The maximum temperature over all nodes and all time steps is taken in (14). In our test instance, we set $\varphi^w := 1800$, $\kappa^e := 0.95$, and $\kappa^w := 0.5$.

4 Results and Conclusions

The objective (1) over the constraint system (2)–(14) together with the integrality constraints on the x and y variables define a mixed-integer linear programming problem. For its solution, we use the numerical solver IBM ILOG CPLEX 12.8.0 (default settings) on a MacBookPro with an Intel Core i7 running 8 threads parallel

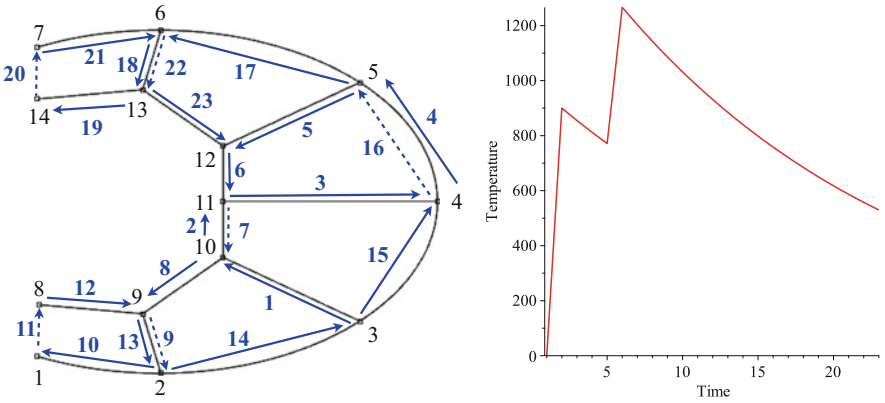


Fig. 2 The trajectory of the welding torch (left). Dashed lines are deadhead transitions, straight lines are welded segments. Temperature of a critical node #11 (right), which is heated in time steps 2 and 6

at 3.1 GHz clockspeed and 16 GB RAM. The input data shown in Fig. 1 give rise to a mixed-integer model having 5397/4813 variables and 2723/2270 constraints with 51,568/45,568 nonzeros before/after presolving. It took 17,812 s to solve this instance to proven global optimality. The computed trajectory of the welding torch is shown in Fig. 2 (left). Node #11 has a critical temperature (i.e., a node where the maximal temperature is reached), its temperature over time is shown in Fig. 2 (right).

In our future work, we extend the problem formulation by not only taking the absolute value of the temperature into account, but also the heat gradient between neighboring positions, which should be limited during production in order to prevent stress forces. A further variant of the problem considers not only a single layer, but several consecutive layers, in order to increase the stability of the component and to avoid certain problems that occur when simply extruding the single-layer solution into the third dimension.

Acknowledgement The authors are grateful to Fabian Gnegel for his helpful comments on an earlier version of the manuscript.

References

1. Edmonds, J., Johnson, E.L.: Matching, Euler tours and the Chinese postman. Math. Program. **5**, 88–124 (1973)
2. Frank, M., Fügenschuh, A., Herty, M., Schewe, L.L.: The coolest path problem. Netw. Heterog. Media **5**(1), 143–162 (2010)
3. Fügenschuh, A., Fügenschuh, M.: Integer linear programming models for topology optimization in sheet metal design. Math. Meth. Oper. Res. **68**(2), 313–331 (2008)

4. Manber, U., Israni, S.: Pierce point minimization and optimal torch path determination in flame cutting. J. Manuf. Syst. **3**(1), 81–89 (1984)
5. Nguyen, L., Buhl, J., Bambach, M.: Decomposition Algorithm for tool path planning for wire-arc additive manufacturing. J. Mach. Eng. **18**(1), 95–106 (2018)
6. Sun, J., Meng, Y., Tan, G.: An integer programming approach for the Chinese postman problem with time-dependent travel time. J. Comb. Opt. **29**, 565–588 (2015)

A Time-Flow Approach for Scheduling and Routing of Fly-in Safari Airplanes

Fabian Gnegel and Armin Fügenschuh

1 Introduction

A company in southern Africa is offering fly-in trips to and in between several locations. Everyday tourist groups have to be picked up at their current location and brought to their next destination. The operating cost highly depends on the distance that the planes cover on their tours, so that an effective routing of the planes is crucial. The routing, however, is constrained by time-windows, limitations on the detour for each tourist group, varying fuel capacities of the planes, the number of seats on the planes, differing weight limits at the airports for takeoff and landing, and the small number of airports with refueling options. Fügenschuh et al. [3] derived a mixed integer linear program (MILP) modeling all these aspects of this problem. They used a time-expanded formulation which required a large number of variables and constraints. The obtained MILP, therefore, could often not be solved even if the fleet was limited to two planes. After identifying that the time constraints were rather soft in most cases, they derived a time-free formulation, a relaxation of the original model, which provided good lower bounds for most instances. In combination with a heuristic for finding primal solutions they were then able to use this relaxation in order to reduce the average optimality gap to less than 5%. We derive a different MILP formulation for the problem, which avoids the (total) time expansion. Boland et al. [1, 2] showed that avoiding the total expansion can have a significant impact on the computation time in similar problems.

Our formulation allows not only for a time-free, but also intermediate time-dependent relaxations. We derive an algorithm that makes use of these relaxations and conduct a computational study on the same test cases as in [3].

F. Gnegel (✉) · A. Fügenschuh
Brandenburg Technical University Cottbus-Senftenberg, Cottbus, Germany
e-mail: gnegel@b-tu.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_42

2 A Time-Flow MILP Formulation of the Problem

Adopting the notations introduced by Fügenschuh et al. [3], $\mathcal{P}$ is used for the set of planes, $\mathcal{V}$ for the set of airports and $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ as the set of direct flights between airports. In order to describe the availability of fuels at the different airports we define the set $\mathcal{F}$, which contains tuples (i, θ), for all available fuel types θ at airports $i \in \mathcal{V}$. The flight requests of the tourist groups are denoted by $\mathcal{R}$. Additionally, we assume that there are given bounds n_i on the number of times an airport $i \in \mathcal{V}$ can be visited (if not given explicitly these can be calculated by taking flight times and the time horizon into account). These bounds are used to define the extended set of airports $\mathcal{V}^{\text{ext}}$ as the set of all tuples $i = (i_1, i_2)$ with $i_1 \in \mathcal{V}$ and $i_2 \in \{1, \ldots, n_{i_1}\}$ and the extended set of trips $\mathcal{A}^{\text{ext}} \subset \mathcal{V}^{\text{ext}} \times \mathcal{V}^{\text{ext}}$, with $(i, j) \in \mathcal{V}^{\text{ext}} \times \mathcal{V}^{\text{ext}}$ iff $(i_1, j_1) \in \mathcal{A}$. For ease of notation all parameters defined for airports in $\mathcal{V}$ and trips in $\mathcal{A}$ may also be indexed by corresponding airports in $\mathcal{V}^{\text{ext}}$ or trips in $\mathcal{A}^{\text{ext}}$ and take the same value. For the plane $p \in \mathcal{P}$ the required fuel type is denoted by ρ_p, the number of passenger seats by $\overline{s}_p$, the airport of the initial departure by D_p, the final airport by A_p, the lower bound on the fuel at departure by $\underline{\varphi}_p$, the upper bound on the fuel at departure by $\overline{\varphi}_p$, and by $\underline{\psi}_p$ and $\overline{\psi}_p$, the respective bounds at arrival. The earliest time for the initial departure of the plane $p \in \mathcal{P}$ is denoted by τ^p_{dep} and the latest time for its last arrival by τ^p_{arr}. For each trip $(i, j) \in \mathcal{A}$ we denote its length by $d_{i,j}$, its cost by $c^p_{i,j}$, and its duration by $\delta^p_{i,j}$, which also depend on the plane $p \in \mathcal{P}$. Furthermore, the respective fuel consumption is given by $\gamma^p_{i,j}$ and the maximal amount of fuel that can be taken by $\overline{f}^p_{i,j}$. For the requests $r \in \mathcal{R}$ we use D_r for the departure airport, A_r for the arrival airport, and s_r for the number of passengers . Furthermore the earliest possible departure time of a request $r \in \mathcal{R}$ is denoted by τ^r_{dep} and the latest arrival time by τ^r_{arr}. Instead of the original time-expanded model of [3] we give a time-flow formulation using the extended set of airports $\mathcal{V}^{\text{ext}}$ as vertices and the extended set of trips $\mathcal{A}^{\text{ext}}$ as arcs.

Variables For the model we introduce the following variables:

$$\forall\, (i, j) \in \mathcal{A}^{\text{ext}},\, p \in \mathcal{P} : \qquad y^p_{i,j} \in \mathbb{N}_0, \tag{1a}$$

$$\forall\, r \in \mathcal{R},\, (i, j) \in \mathcal{A}^{\text{ext}},\, p \in \mathcal{P} : \qquad x^{r,p}_{i,j} \in \{0, 1\}, \tag{1b}$$

$$\forall\, r \in \mathcal{R},\, p \in \mathcal{P},\, i \in \mathcal{V}^{\text{ext}} \text{s.t. } i_1 = D_r : \qquad x^{i,r,p}_{\text{dep}} \in \{0, 1\}, \tag{1c}$$

$$\forall\, r \in \mathcal{R},\, p \in \mathcal{P},\, i \in \mathcal{V}^{\text{ext}} \text{s.t. } i_1 = A_r : \qquad x^{i,r,p}_{\text{arr}} \in \{0, 1\}, \tag{1d}$$

$$\forall\,(i,j)\in\mathcal{A}^{\text{ext}},\ p\in\mathcal{P}: \qquad f^p_{i,j}\in\mathbb{R}_+, \tag{1e}$$

$$\forall\,p\in\mathcal{P}: \qquad f^p_{\text{dep}}\in\left[\underline{\varphi}_p,\overline{\varphi}_p\right],\ f^p_{\text{arr}}\in\left[\underline{\psi}_p,\overline{\psi}_p\right], \tag{1f}$$

$$\forall\,(i,j)\in\mathcal{A}^{\text{ext}},\ p\in\mathcal{P}: \qquad t^p_{i,j}\in\mathbb{R}^+. \tag{1g}$$

The variables in (1a) are used to count the number of times a trip (i,j) is flown by plane p. Those in (1b) are used analogously for the routing of requests. With the additional variables in (1c), which indicate at which copy the request departs, and the ones in (1d) for the copy where the route of the request finishes. The fuel at takeoff of plane $p\in\mathcal{P}$ for its trips from airport i to airport j is given by the variables in (1e) and the initial amount of fuel and remaining fuel are given by the variables in (1f). Finally, the time of arrival of a trip $(i,j)\in\mathcal{A}^{\text{ext}}$ is given by the variables in (1g).

Objective The objective is to minimize the operative cost, which only depends on the routing of the planes:

$$\sum_{(i,j)\in\mathcal{A}^{\text{ext}}}\sum_{p\in\mathcal{P}} c^p_{i,j}\cdot y^p_{i,j}\to\min. \tag{2}$$

Constraints The following types of constraints are used in the MILP:

$$\begin{aligned}\forall\,j\in\mathcal{V}^{\text{ext}},\ p\in\mathcal{P}: &\sum_{i:(i,j)\in\mathcal{A}^{\text{ext}}} y^p_{i,j}+\begin{cases}1,\ \text{if } j=D_p \text{ and } j_2=1,\\ 0,\ \text{else},\end{cases}\\ =&\sum_{k:(j,k)\in\mathcal{A}^{\text{ext}}} y^p_{j,k}+\begin{cases}1,\ \text{if } j=A_p \text{ and } j_2=1,\\ 0,\ \text{else}.\end{cases}\end{aligned} \tag{3a}$$

$$\begin{aligned}\forall\,j\in\mathcal{V}^{\text{ext}},\ r\in\mathcal{R},\ p\in\mathcal{P}: &\sum_{i:(i,j)\in\mathcal{A}^{\text{ext}}} x^{r,p}_{i,j}+\begin{cases}x^{i,r,p}_{\text{dep}},\ \text{if } j=D_r,\\ 0,\qquad \text{else},\end{cases}\\ =&\sum_{k:(j,k)\in\mathcal{A}^{\text{ext}}} x^{r,p}_{j,k}+\begin{cases}x^{i,r,p}_{\text{dep}},\ \text{if } j=A_r,\\ 0,\qquad \text{else},\end{cases}\end{aligned} \tag{3b}$$

$$\forall\, j \in \mathcal{V}^{\text{ext}},\, p \in \mathcal{P}: \sum_{i:(i,j)\in\mathcal{A}^{\text{ext}}} f^p_{i,j} + \begin{cases} f^p_{\text{dep}}, \text{ if } j = D_p \text{ and } j_2 = 1, \\ 0, \quad \text{else}, \end{cases}$$
$$= (\leq) \sum_{k:(j,k)\in\mathcal{A}^{\text{ext}}} \left(f^p_{j,k} + \gamma^p_{j,k} \cdot y^p_{j,k} \right) + \begin{cases} f^p_{\text{arr}}, \text{ if } j = A_p \text{ and } j_2 = 1, \\ 0, \quad \text{else}, \end{cases} \tag{3c}$$

$$\forall\, j \in \mathcal{V}^{\text{ext}},\, p \in \mathcal{P}: \sum_{k:(j,k)\in\mathcal{A}^{\text{ext}}} t^p_{j,k} + \begin{cases} \tau^p_{\text{arr}}, \text{ if } j = A_p \text{ and } j_2 = 1, \\ 0, \quad \text{else}, \end{cases}$$
$$\geq \sum_{i:(i,j)\in\mathcal{A}^{\text{ext}}} \left(t^p_{i,j} + \delta^p_{i,j} \cdot y^p_{i,j} \right) + \begin{cases} \tau^p_{\text{dep}}, \text{ if } j = D_p \text{ and } j_2 = 1, \\ 0, \quad \text{else}, \end{cases} \tag{3d}$$

$$\forall\, (i,j) \in \mathcal{A}^{\text{ext}},\, p \in \mathcal{P}: \sum_{r\in\mathcal{R}:(i,j)\in\mathcal{A}^{\text{ext}}} s_r \cdot x^{r,p}_{i,j} \leq \overline{s}_p \cdot y^p_{i,j}, \tag{3e}$$

$$\forall\, (i,j) \in \mathcal{A}^{\text{ext}},\, p \in \mathcal{P}: f^p_{i,j} \leq \overline{f}^p_{i,j} \cdot y^p_{i,j}, \tag{3f}$$

$$\forall\, p \in \mathcal{P},\, (i,j) \in \mathcal{A}^{\text{ext}},\, r \in \mathcal{R}: t^p_{i,j} \leq \tau^r_{\text{arr}} + 2M(1 - x^{r,p}_{i,j}) + M(y^p_{i,j} - 1), \tag{3g}$$

$$\forall\, p \in \mathcal{P},\, (i,j) \in \mathcal{A}^{\text{ext}},\, r \in \mathcal{R}: t^p_{i,j} \geq \tau^r_{\text{dep}} - 2M(1 - x^{r,p}_{i,j}) - M(y^p_{i,j} - 1), \tag{3h}$$

$$\forall\, p \in \mathcal{P},\, j \in \mathcal{V}^{\text{ext}} \text{s.t. } j_2 \neq n_{j_1}: \sum_{i:(i,j)\in\mathcal{A}^{\text{ext}}} y^p_{i,j} \leq 1, \tag{3i}$$

$$\forall\, p \in \mathcal{P},\, j \in \mathcal{V}^{\text{ext}} \text{s.t. } j_2 \neq n_{j_1}: \sum_{k:(j,k)\in\mathcal{A}^{\text{ext}}} y^p_{j,k} \leq 1, \tag{3j}$$

$$\forall\, p \in \mathcal{P},\, j \in \mathcal{V}^{\text{ext}} \text{s.t. } j_2 \neq n_{j_1}: \sum_{i:(i,j)\in\mathcal{A}^{\text{ext}}} y^p_{i,j} \leq \sum_{i:(i,j)\in\mathcal{A}^{\text{ext}}} y^p_{i,(j_1,j_2+1)}. \tag{3k}$$

Constraints of similar structure are used for the routes of the planes, (3a), the routes of the requests, (3b), tracking the fuel consumption of the trips, (3c), and tracking the departure times, (3d). All of these are flow conservation constraints that in combination with the coupling constraints (3) and (3f) ensure that the variables take values according to the problem description. Such a use of flow variables for tracking traveling times was introduced by Gavish and Graves [4]. The flow conversation laws for the fuel variables (3c) were already used by Fügenschuh et al. [3] and we note that the inequality in the parentheses is used iff the required fuel type of a plane is available at the respective airport. For the time windows of the requests big-M type constraints are used in the constraints (3g) and (3h). Finally

constraints (3i) and (3j) ensure that each vertex can only be visited once, and (3k) ensures that copies with lower index take priority for visits. As they are not relevant for our methods, we refer for the constraints on the payloads and detours to [3].

3 Relaxations of the MILP

Fügenschuh et al. [3] have successfully used a relaxation of their time-expanded formulation, referred to as the time-free model, to reduce the optimality gaps and computation time on the benchmark instances. By setting $n_i = 1$ for all $i \in \mathcal{V}$ and removing constraints (3d), (3g), and (3h) an identical relaxation can be obtained from our model. In fact, any reduction of the number of copies of any airport gives a relaxation to the time-flow formulation. Furthermore, if the optimal routing given by a relaxation is feasible for the full model it is also proven optimal. As shown by Fügenschuh et al. the relaxed MILP are much easier to solve (often in seconds) than the expanded formulations (which often remain unsolved after hours), which raises the question whether intermediate graphs are also useful relaxations. In order to explore the benefits of these relaxations we formulate Algorithm 1 which has the goal to construct intermediate graphs efficiently based on previously checked relaxations with less copies of airports. Algorithm 1 starts with one copy for each airport. If the obtained schedule is already feasible for the full model, the optimal solution has been found. Otherwise a loop is entered that increases the number of copies of only those airports that are visited more often than the number of current copies in the reduction. If the routing of a relaxation only visits each copy at most once, no cycles are included, and feasibility to the full model is ensured. Therefore

Algorithm 1: Iterative construction of relaxations

Data: a full parameter description of the airplane routing problem, infeasible *CurrentSolution*

for $i \in \mathcal{V}$ **do**
  $\tilde{n}_i \leftarrow 1$;
while *CurrentSolution is infeasible for Time-Flow MILP* **do**
  CurrentReduction $\leftarrow$ Relaxation of the Time-Flow MILP with $n_i = \tilde{n}_i$ for all $i \in \mathcal{V}$;
  Solve *CurrentReduction*;
  CurrentSolution $\leftarrow$ Optimal solution of *CurrentReduction*;
  for $i \in \mathcal{V}$ **do**
    stops $\leftarrow$ the maximum of the total number of arrivals and departures from i in *CurrentSolution*;
    if $\tilde{n}_i < stops$ **then**
      $\tilde{n}_i \leftarrow$ the total number of visits to i;
OptimalSolution $\leftarrow$ *CurrentSolution*

the number of copies of at least one airport is increased in each iteration and the loop terminates in finitely many iterations. We note that for the computational studies Algorithm 1 is tweaked in such a way that all other infeasible solutions found during the branch-and-bound process are also abrogated by introducing additional copies.

4 Computational Results

A computational study of the new model and the related relaxations is used to asses performance on the benchmark instances from [3] on a MacBook Pro with 16 GB RAM and a 3.1 Ghz Intel Core i7 processor. For the solution of the MILP the state-of-the-art solver Cplex 12.7.1 was used. The results and for comparison also the results of the time-free approach from [3] can be found in Table 1. They indicate that the time-flow approach using Algorithm 1 outperforms the time-free approach on several levels. Not only is the average computation time almost halved, the optimality gap is also reduced to less than a half. Furthermore, more instances have been solved to optimality and the objective value of the best known solution was reduced for some instances. We conclude that the stronger relaxations given by the MILP derived in Algorithm 1 can be used to reduce computation times and find solutions of higher quality, although the price of solving on average 3.21 MILP has to be paid.

Table 1 Comparison of computational results

	b&c time	b&c dual bd.	Objective value	Gap	Loop iteration
Instance	Algorithm 1/[3]	Algorithm 1/[3]	Algorithm 1/[3]	Algorithm 1/[3]	Algorithm 1
BUF-ANT	10,800/10,800	17,760/16,346	20,591/20,724	13.75/21.12%	3
BUF-BEE	10,800/10,800	16,687/16,251	17,456/17,633	4.41/7.84%	3
BUF-EGL	10,800/10,800	21,150/19,738	21,575/22,113	1.97/10.74%	3
BUF-GNU	10,800/10,800	17,200/16,763	17,350/17,350	0.86/3.39%	6
BUF-JKL	10,800/10,800	18,849/19,138	20,774/20,774	9.27/7.87%	2
BUF-LEO	10,800/10,800	22,750/22,488	24,938/24,938	8.77/9.82%	2
EGL-BEE	135/10,800	16,652/16,447	16,653/16,653	0.0/1.23%	5
EGL-GNU	45/10,800	19,225/16,825	19,225/20,238	0.0/16.86%	4
EGL-LEO	10/8280	19,388/19,388	19,388/19,388	0.0/0.0%	2
GNU-BEE	2195/10,800	11,165/10,282	11,165/11,311	0.0/9.09%	3
GNU-JKL	898/10,800	11,098/10,244	11,098/11,098	0.0/7.70%	3
GNU-LEO	29/8928	17,863/17,863	17,863/17,863	0.0/0.0%	3
LEO-AIV	1/12	13,615/13,615	13,615/13,615	0.0/0.0%	1
LEO-BOK	20/5664	15,372/15,372	15,372/15,372	0.0/0.0%	5
Average	4798/9446	17,055/16,438	17,673/17,790	3.4/7.5%	3.21

Acknowledgement This project was funded by the German Research Association (DFG), grant number FU 860/1-1.

References

1. Boland, N., Hewitt, M., Vu, D.M., Savelsbergh, M.: Solving the traveling salesman problem with time windows using time-expanded networks. In: International Conference on AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems, pp. 254–262. Springer, Cham (2017)
2. Boland, N., Hewitt, M., Marshall, L., Savelsbergh, M.: The continuous-time service network design problem. Oper. Res. **65**(5), 1303–1321 (2017)
3. Fügenschuh, A., Nemhauser, G., Zeng, Y.: Scheduling and Routing of Fly-in Safari Planes Using a Flow-over-Flow Model, pp. 419–447. Springer, Berlin (2013)
4. Gavish, B., Graves, S.C.: The Traveling Salesman Problem and Related Problems. Technical Report OR 078-78, Operations Research Center, Massachusetts Institute of Technology, 1978

Designing a Water Supply Network for Slums in Rio de Janeiro Using Mixed-Integer Programming

Marvin Meck, Lea Rausch, John Friesen, Michael Wurm, Hannes Taubenböck, Lena Altherr, and Peter F. Pelz

1 Introduction

The rapid past and on-going urbanisation and growth of cities is observed especially in developing countries [1], leading to a massive strain on the infrastructure of these cities and, e.g., to underdeveloped water supply systems [2]. According to the United Nations *World Cities Report 2016*, approximately 663 million people worldwide do not have access to sufficient amounts of water while at least 1.8 billion people have to rely on a source of drinking water that is fecally contaminated [3]. This issue is especially prominent in areas with people living in informal settlements or slums [4]. The goal of this paper is to develop strategies for improving the living conditions of the poor by using the spatial patterns within the cities as input for an optimisation. We introduce a holistic approach to design an economically optimal water supply system for slums in urban areas. Based on remote sensing data, methods of mathematical optimisation and graph theory are used to find an optimal water supply network. The model combines different approaches using various motorised vehicles as well as pipe systems for water supply and is formulated as a mixed-integer problem (MIP). In this paper, a previously presented model [5, 6] is modified and applied to a slum cluster in Rio de Janeiro, Brazil. The spatial

M. Meck · L. Rausch · J. Friesen · L. Altherr · P. F. Pelz (✉)
Department of Mechanical Engineering, Technische Universität Darmstadt, Darmstadt, Germany
e-mail: marvin.meck@fst.tu-darmstadt.de; lea.rausch@fst.tu-darmstadt.de; john.friesen@fst.tu-darmstadt.de; lena.altherr@fst.tu-darmstadt.de; peter.pelz@fst.tu-darmstadt.de

M. Wurm · H. Taubenböck
Earth Observation Center (EOC), Deutsches Zentrum für Luft- und Raumfahrt e.V. (DLR), Weßling, Germany
e-mail: michael.wurm@dlr.de; hannes.taubenboeck@dlr.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_43

knowledge on locations and sizes of morphologic slums relies on the empirical conceptualisation of such structures [7] and their classifications [8].

2 Remote Sensing-Based Classification of Morphologic Slums

The built environment can be an expression of inequality in cities visible from space [9]. The developed approaches for spatially capturing these inequalities using remote sensing data rely on physical features characterising the built environment. Indicators such as organic, complex pattern, high densities, or small building sizes are applied [8, 10]. Wurm and Taubenböck [7] prove that these morphologic slum classifications allow to localise the urban poor with high accuracies [7]. This paper relies on slum classifications on block level delimiting these areas from formal settlements within Rio de Janeiro.

3 Modelling Water Supply Design as MIP

The presented approach aims at designing the water supply infrastructure for all slums within one large city, providing each slum with the required amount of water based on its size. The daily water need for each slum i can be calculated as the product of slum size, estimated population density and the daily need per person. The current model uses one water delivery point (also referred to as *water works*) in each city, the location of which has to be provided as input data. One or multiple types of the following connections can be chosen between any of the slums or water delivery point and any slum: *Firstly*, water can be transported via pipes with different diameters. Currently, three different sizes are defined to choose from, the use is limited to a single pipe for a given connection. *Secondly*, two different types of motorised vehicles combined with three different sized water tanks for storage within the slum can be used. The optimal solution for the supply network is characterised by the minimal total costs over a specified period of time, including investment and operating costs. To account for flow conditions and capacity restrictions, various additional requirements are modeled in the MIP. For further details, we refer to [5, 6].

Objective Function The objective is to minimise the total costs including investment (fix) and operating costs (op) within a specified time frame. Operating costs scale exponentially with the volume flow and a constant. In this context, we define the variable costs $\mathrm{Cost}_{\mathrm{P,\,var}}$ and the exponential volume flow Q_{exp}. This relation is

Table 1 Decision variables and parameters of the mixed-integer linear problem

Variable	Description
$x_{\text{type}}(i, j, k)$	Binary indicator if a pipe/ truck/ tank of type k is chosen on (i, j)
$Q_{\text{exp}}(i, j)$	Linearised volume flow to the power of 2.75
Parameter	Description
N	Set of all slums, $N \subset \mathbb{N}$
N_{w}	Set of slums and water works w, $N_{\text{w}} = N \cup \{w\}$
$N_{\text{truck}}^{\max}$	Maximal number of trucks allowed between two slums
K_{type}	Set of available pipe/ truck/ tank types
$\text{Cost}_{\text{pipe, fix}}(i, j, k)$	Fixed pipe costs of type $k \in K_{\text{pipe}}$ for distance from slums i to j
$\text{Cost}_{\text{pipe, var}}(i, j, k)$	Variable pipe cost factor of type $k \in K_{\text{pipe}}$
$\text{Cost}_{\text{truck}}(i, j, k)$	Truck cost of type $k \in K_{\text{truck}}$ for distance from i to j
$\text{Cost}_{\text{tank}}(k)$	Tank cost for type $k \in K_{\text{tank}}$

further discussed in Sect. 4. With the parameters and variables described in Table 1, the objective function can be written as:

$$
\begin{aligned}
min \sum_{i \in N_{\text{w}}} \sum_{j \in N} \sum_{k \in K_{truck}} & x_{\text{truck}}(i, j, k) \cdot \text{Cost}_{\text{truck}}(i, j, k) \\
& + \sum_{k \in K_P} x_{\text{pipe}}(i, j, k) \cdot \text{Cost}_{\text{P, fix}}(i, j, k) + Q_{\text{exp}}(i, j) \cdot \text{Cost}_{\text{P, var}}(i, j, k) \\
& + \sum_{i \in N} \sum_{k \in K_{tank}} x_{\text{tank}}(i, k) \cdot \text{Cost}_{\text{tank}}(k)
\end{aligned}
$$

Constraints Besides meeting the daily need, additional constraints ensure the full functionality of the water supply system and reasonable solutions. The underlying requirements are summarised in the following. For a detailed formulation of the equations, see [5] and [6]. The constraints can be grouped into four different categories: The capacity of a connection needs to exceed the volume flow (1), the total volume flow is the sum of the flow via pipes and trucks. The total incoming volume flow needs to be equal to the sum of outgoing flow and daily need for each slum (2). Pipes and trucks can only be chosen if the connection is used. The number of trucks per edge is limited by $N_{\text{truck}}^{\max}$ and only one pipe is allowed. In case a vehicle is used to carry water into or out of the slum, a tank is required for storage in the slum (3). Geographic barriers (input information for individual connections), like rivers, prevent specific connections (4).

4 Enhancement and Modifications of the MIP

As shown in Sect. 4, the total costs account for investment and operating costs of the pipes, vehicles and tanks used to provide sufficient water supply. Modelling of the cost terms therefore has a major impact on quality and validity of the model. Investment costs are estimated using empirical data and correlation models. Operating costs for trucks include fuel and labour costs. Hence, they are a function of time or, assuming a constant average velocity, distance travelled. Operating costs for pipes have to account for electrical energy needed to pump water overcoming the pressure drop due to frictional losses. The estimate for operating costs of pipes has been improved to yield better model quality and is described in the following. In engineering it is common practice to make use of stream filament theory in applications where the whole region of interest can be thought of as one streamtube. This is the case e.g., for flows in tubes, through nozzles and diffusers. Even flow between the blades of turbomachines, like pumps, can be approximated in this manner. Within the framework of this assumption, flow quantities are assumed constant over the cross-section and are only functions of the arc length along the streamline. Though the assumption of constant flow variables over the cross-section requires frictional effects to be neglected, those losses can be discussed phenomenologically by introducing them as additional pressure drop Δp_l [11]. For turbulent pipe flows with hydraulically smooth surfaces, the pressure drop can be written as

$$\Delta p_l = \zeta \varrho \, u^2/2 \qquad \text{with} \quad \zeta = \lambda \frac{L}{d_h} = \frac{0.3146}{\mathrm{Re}^{0.25}} \frac{L}{d_h} \quad \text{and} \quad \mathrm{Re} := \frac{\varrho \, u \, d}{\mu}$$

with the density of the flow medium ϱ, the flow velocity u and the loss factor ζ. Blasius gives a simple correlation to estimate the loss factor ζ depending on the Reynolds number Re, the pipe length L, the dynamic viscosity of the flow medium μ and the hydraulic diameter d_h, which, for fully flooded pipes, is equal to the pipe diameter d [12]. The cost term can now be calculated by the dissipated energy $P_{\mathrm{diss}} = Q \, \Delta p_l$, pump efficiency η_{Pump} and costs for electrical energy $\mathrm{Cost}_{\mathrm{elec}}$. Since the volume flow $Q(i, j)$ between slums i and j is used as the decision variable, the costs are written as a function of volume flow $Q = u \, A$ with the cross sectional area $A = \pi/4 \, d^2$ for a circular pipe. By this, one yields the operational costs for a pipe depending on type (diameter), length and transported volume flow:

$$\mathrm{Cost}_{\mathrm{P,op}} = \frac{1}{\eta_{\mathrm{Pump}}} \mathrm{Cost}_{\mathrm{elec}} \, P_{\mathrm{diss}} = 0.3146 \, \frac{\mathrm{Cost}_{\mathrm{elec}}}{\eta_{\mathrm{Pump}}} \sqrt[4]{\frac{2^{10} \varrho^3 \mu L^4}{\pi^7 d^{19}}} \cdot Q^{2.75}$$

5 Results and Conclusion

Results Input data was generated by classification of morphologic slums relying on very high resolution optical satellite data and the related empirical conceptualisation of underserved areas. From an engineering perspective, the choice of reasonable pipe diameters appears to be particularly important. Considering higher volume flows and reduced frictional losses, larger pipes will have to be chosen to transport water between multiple clusters (and/ or over long distances). On the other hand, smaller diameters would be preferred for supply in a cluster itself, to reduce costs.

In theory, this problem could be eliminated by allowing for a wider range of different sized pipes to chose from. This would, however inevitably lead to an increase of the already high computational costs of the model. Following this train of thought, we compare the impact of varying the choice of pipe sizes on the solution as well as on the computational effort required to derive it. To solve the problem, the commercial optimisation software CPLEX, version 12.8 was used. The MIP was implemented via IBM ILOG CONCERT language in C++ and solved on a dedicated machine using regular workstation hardware (CPU: 3.6 GHz; Memory: 64 GB RAM). Unlike in earlier implementations, the necessary linearisation of non-linear functions, e.g., the above mentioned exponential volume flow for the pipes, is now achieved by making use of the IBM ILOG CONCERT built-in functionality ILOPIECEWISELINEAR, vastly decreasing the number of variables and increasing performance. It has to be noted that, though the mathematical model is not changed, transparency is lost. However, with both linearisation methods leading to the same results, in our instances, the loss of information can be justified by the reduced computational effort. The total costs for three different scenarios, all over a period of five years, were calculated. The resulting networks are shown in Figs. 1, 2, and 3.

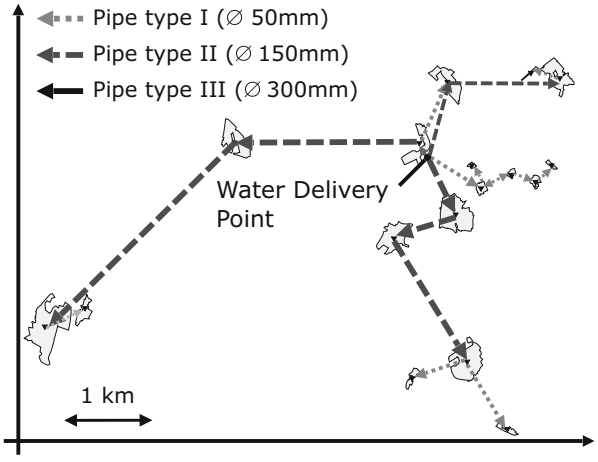


Fig. 1 Solution for small pipes

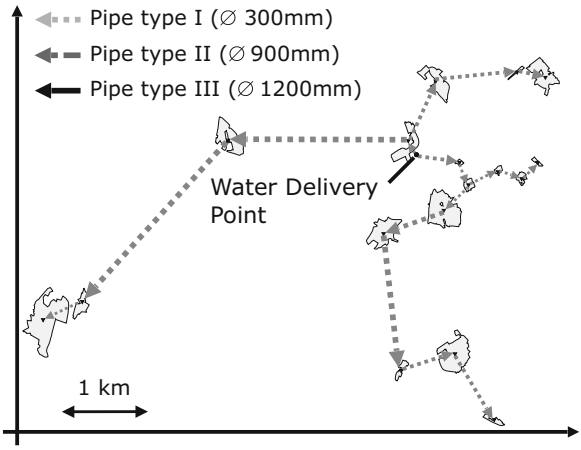


Fig. 2 Solution for large pipes

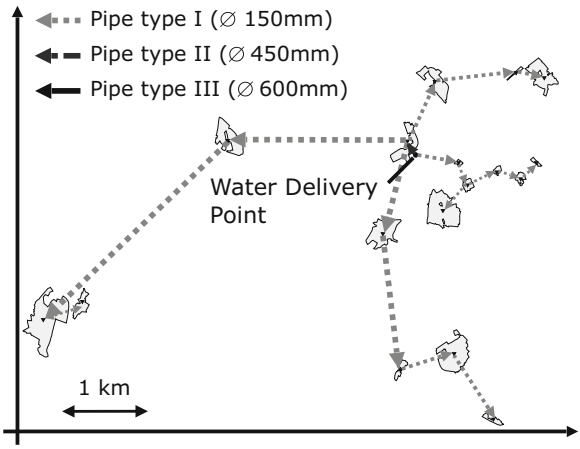


Fig. 3 Solution for med. pipes

Common pipe diameters typically range anywhere from 25 to 150 mm [13] and can provide low to medium high volume flows. If larger flow rates are required, usually concrete pipes are used. Here, diameters from 300 up to 4000 mm and higher are available. The results show that in this particular example, a selection for pipe diameters ranging around 150 mm seems to provide solutions with the overall lowest costs. The analysis also showed that the variation of input parameters can have a strong effect on memory demand and computational effort, as shown in Table 2. Making use of "best-estimate" node selection and "strong branching" variable selection strategies, excessive memory consumption could be avoided.

Table 2 Memory demand and computation times for different diameter sets

Solution strategy	50/150/300 mm	150/450/600 mm	300/600/1200 mm
Default	89 GB, 15 h	1.3 GB, 2:30 min	1.3 GB, 3:20 min
Best-estimate search[a]	233 MB, 10:30 min	1.7 GB, 10 min	420 MB, 10 min

[a]incl. strong branching, allowing compressed nodefiles if memory demand exceeds system memory

Conclusion Although performance was increased compared to earlier versions of the model [5], due to memory restrictions it is currently still impossible to find a global solution in the order of magnitude necessary to solve an entire megacity like Rio de Janeiro. With the current model, we are able to solve clusters limited to about $N_W = 30$. In this case, depending on the input variables chosen, solution trees can be as large as 180 GB even with memory efficient solution strategies. One approach to work around this issue is to cluster the slums, splitting the master problem into various sub-problems, which are then solved individually. This approach is also promising in regards to the issue of finding the best suited range of pipe sizes. It allows for solving larger instances by saving computational resources, although note that global optimality is lost.

References

1. Taubenböck, H., Esch, T., Felbier, A., Wiesner, M., Roth, A., Dech, S.: Monitoring of mega cities from space. Remote Sens. Environ. **117**, 162–176 (2012)
2. Van der Bruggen, B., Borghgraef, K., Vinckier, C.: Causes of water supply problems in urbanised regions in developing countries. Water Resour. Manag. **24**(9), 1885–1902 (2010)
3. United Nations Human Settlements Programme: World Cities Report 2016 (2016)
4. Kraas, F., et al.: Der Umzug der Menschheit: Die transformative Kraft der Städte. WBGU—German Advisory Council on Global Change, Berlin (2016)
5. Rausch, L., Friesen, J., Altherr, L.C., Pelz, P.F.: Using mixed-integer programming for the optimal design of water supply networks for slums. In: Operations Research Proceedings 2017: Selected Papers of the Annual International Conference of the German Operations Research Society (GOR). Springer, Berlin (2017)
6. Rausch, L., Friesen, J., Altherr, L.C., Meck, M., Pelz, P.F.: A holistic concept to design optimal water supply infrastructures for informal settlements using remote sensing data. Remote Sens. **10**(2), 216 (2018)
7. Wurm, M., Taubenböck, H.: Detecting social groups from space–remote sensing-based mapping of morphological slums and assessment with income data. Remote Sens. Lett. **9**(1), 41–50 (2018)
8. Taubenböck, H., Kraff, N.J., Wurm, M.: The morphology of the arrival city: a global categorization based on literature surveys and remotely sensed data. Appl. Geogr. **92**, 150–167 (2018)
9. Sliuzas, R., Mboup, G., de Sherbinin, A.: Report of the expert group meeting on slum identification and mapping (2008)
10. Kohli, D., Sliuzas, R., Kerle, N., Stein, A.: An ontology of slums for image – based classification. Comput. Environ. Urban. Syst. **36**(2), 154–163 (2012)

11. Spurk, J., Aksel, N.: Strömungslehre: Einführung in die Theorie der Strömungen. Springer-Lehrbuch. Springer, Berlin (2010)
12. Bird, R.B., Stewart, W.E., Lightfoot, E.N.: Transport Phenomena, 2nd edn. Wiley, London (2001)
13. DIN EN 10255: Non-Alloy steel tubes suitable for welding and threading - Technical delivery conditions. Standard, Deutsches Institut Für Normung (2007)

Optimizing Pressure Screen Systems in Paper Recycling: Optimal System Layout, Component Selection and Operation

Tim M. Müller, Lena C. Altherr, Marja Ahola, Samuel Schabel, and Peter F. Pelz

1 Introduction

With a worldwide utilization rate of 59%, recovered paper is the most important raw material in paper industry [7]. Adhesive contaminants in this raw material, so-called stickies, reduce the paper quality considerably. During the recycling process, as many stickies as possible are separated from valuable fibers using multi-stage screening systems. When planning such systems, one has to select suitable screens, determine their interconnection, as well as operational parameters. Moreover, conflicting objectives, such as maximizing quality while maximizing fiber yield and minimizing energy consumption, have to be considered.

In [3], a Mixed-Integer Nonlinear Program (MINLP) was introduced to optimize the interconnection, as well as the operational parameters. In this contribution, we extend this model: (1) To assess the energy consumption of the system, we model the volume flow of the fiber suspensions. (2) We consider more system layouts by allowing to split connections between different screens. (3) To not only optimize the screen interconnection, but also their individual designs, we integrate a generic screen model. (4) Besides quality and fiber yield, we introduce the resource-consumption and complexity of the system as additional objectives.

T. M. Müller (✉) · L. C. Altherr · P. F. Pelz
Chair of Fluid Systems, Technische Universität Darmstadt, Darmstadt, Germany
e-mail: tim.mueller@fst.tu-darmstadt.de; lena.altherr@fst.tu-darmstadt.de; peter.pelz@fst.tu-darmstadt.de

M. Ahola · S. Schabel
Chair of Paper Technology and Mechanical Process Engineering, Technische Universität Darmstadt, Darmstadt, Germany
e-mail: ahola@papier.tu-darmstadt.de; schabel@papier.tu-darmstadt.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_44

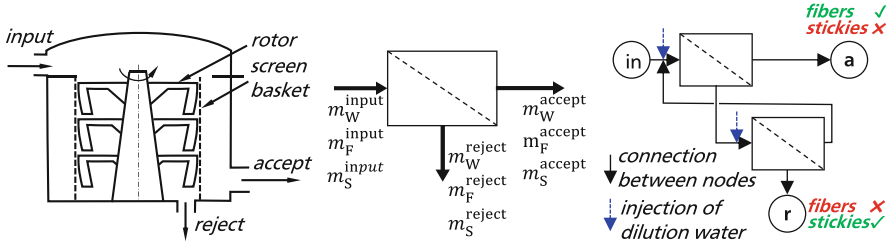


Fig. 1 *Left:* Single pressure screen. *Middle:* Abstracted pressure screen *Right:* Exemplary system using two screens and a cascade feedback layout

2 Technical Application

Pressure screens, cf. Fig. 1, are used to separate valuable fibers from contaminative stickies within the fiber suspension. Each screen has one input and two output connections (accept and reject). For separating the components, a slotted or holed screen basket is used. Since the fibers' probability to pass the screen is higher than the stickies', fibers accumulate in the accept (high quality stream), and stickies in the reject (low quality stream). In a fine screening system, several pressure screens are interconnected, cf. Fig. 1, right. The output of one screen is fed into another screen which increases the separation efficiency.

We model the idealized suspension as a fiber massflow, sticky surface flow and water volume flow represented by $m_{\{\mathrm{F,S,W}\}}$, respectively. By considering the water volume flow, compared to [3] it is possible to (1) use the volumetric instead of the mass reject rate, which is the more practical operational parameter, (2) calculate and limit the consistency (ratio of fiber mass flow to volume flow), so that a blinding of the screens can be avoided, (3) incorporate dilution water into the model, and (4) derive an indirect measure for the energy consumption by summing up the volume flow through all screens.

3 Mixed-Integer Nonlinear Optimization Problem

The resulting MINLP is given in (1)–(23). Small letters indicate variables, capital and Greek letters indicate parameters, cf. Table 1. To model the system's layout, two directed graphs G^{λ} are used, one representing the accept connections ($\lambda = \mathrm{acc}$), one the reject connections ($\lambda = \mathrm{rej}$). Their nodes comprise the screens $Sc = \{sc_1, \ldots, sc_{N_{\mathrm{sc}}}\}$, the system input in, the system total accept a and reject r. We denote source nodes by $V^+ = Sc \cup \{in\}$ and sink nodes by $V^- = Sc \cup \{a, r\}$. Adjacency matrices $t^{\lambda}_{i,j}$ indicate whether an accept/reject connection from source node $i \in V^+$ to sink node $j \in V^-$ exists or not. The flow of component $k \in K = \{\mathrm{F, S, W}\}$ on edge (i, j) is given by flow matrices $m^{\lambda}_{i,j,k}$.

Table 1 Decision variables, sets and parameters

Variable	Domain	Description
$t_{i,j}^{\lambda}$	$\{0, 1\}$	Connection indicator between nodes i and j, $\lambda \in \Lambda$
$m_{j,k}^{\text{feed}}$	$\mathbb{R}_0^+$	Feed flow of component $k \in K$ into node $j \in V^-$
$m_{i,j,k}^{\lambda}$	$\mathbb{R}_0^+$	Flow of component $k \in K$ between accept or reject of node $i \in V^+$ and feed of node $j \in V^-$, $\lambda \in \Lambda$
$m_{i,\text{W}}^{\text{dil}}$	$\mathbb{R}_0^+$	Flow of dilution water into node $i \in V^+$
r_{sc}	[0.1, 0.8]	Volumetric reject ratio of screen $sc \in Sc$
$p_{sc,k}$	$\mathbb{R}_0^+$	Passage ratio of screen $sc \in Sc$ for component $k \in \{\text{F, S}\}$
$b_{sc,d}$	$\{0, 1\}$	Indicator for chosen design $d \in D$ for screen $sc \in$ Sc
$w_{i,j}^{\lambda}$	$\mathbb{N}_0^+$	Share of flow through connection (i, j) with $\lambda \in \Lambda$
i_l	$\mathbb{R}_0^+$	Performance indicators with $l \in \{\text{F, S, W, E}\}$
Set		Description
Sc		Set of all screens
$\{in, a, r\}$		Set of system input, accept and reject node
V^+		Set of nodes with a possible outgoing connection $V^+ = Sc \cup \{\text{in}\}$
V^-		Set of nodes with a possible incoming connection $V^- = Sc \cup \{\text{a, r}\}$
K		Set of modeled components {F, S, W} (fibers, stickies, water)
D		Set of possible screen designs
Λ		Index set {accept, reject}
Parameter	Range	Description
$I_l^{\text{max/min}}$	$\mathbb{R}_0^+$	Bounds of performance indicators with $l \in \{\text{F, W, E}\}$
N_{sc}	$\mathbb{N}_0^+$	Maximum number of screens used in the system
$N_{\text{split}}^{\text{max}}$	$\mathbb{N}_0^+$	Maximum number of splitting connections
M_k^{input}	$\mathbb{R}_0^+$	Total input flow of component $k \in K$ into the system
M_k^{max}	$\mathbb{R}_0^+$	Maximum pipe capacity of component $k \in K$
$P_{k,d}^{\text{pre}}$	$\mathbb{R}_0^+$	Passage ratio of fibers and stickies ($k \in \{\text{F, S}\}$) for design $d \in D$
C^{max}	0.04	Maximum consistency (ratio of fiber mass to water)
W^{inkr}	$\mathbb{N}^+$	Parameter for calculating splitting ratio of flow

We use the ϵ-constraint method to deal with the multi-criteria problem. We maximize the relative sticky reduction i_{S} in (1). The fiber yield i_{F}, the energy indicator i_{E}, the relative amount of dilution water i_{W} and the number of splitting connections are bounded in (2)–(5). Mass conservation holds for all components. Neglecting aggregation and disintegration of stickies, and the compressibility of water, we assume conservation of surface flow and volume. Each sink node's feed flow is calculated in (6) and (7). (8) prevents an injection of dilution water in the system's outputs. The flow conservation is given by (9). Flow between nodes is

possible if a connection exists, cf. (10). Each screen's input flow is divided between reject and accept. The ratio of the waterflow is defined by the volumetric reject rate, which can be set during operation, cf. (11). The distribution of fibers and stickies in (12) depends on the volumetric reject rate and the passage ratio [6] (probability of a single fiber P_F or sticky P_S to pass the screens' basket).

$$\text{max} \qquad i_S = \frac{M_\mathrm{S}^\mathrm{input} - m_\mathrm{a,S}^\mathrm{feed}}{M_\mathrm{S}^\mathrm{input}} \tag{1}$$

subject to

$$i_\mathrm{F} = \frac{m_\mathrm{a,F}^\mathrm{feed}}{M_\mathrm{F}^\mathrm{input}} \geq I_\mathrm{F}^\mathrm{min} \tag{2}$$

$$i_\mathrm{E} = \frac{\sum_{sc \in Sc} m_{sc,\mathrm{W}}^\mathrm{feed}}{M_\mathrm{W}^\mathrm{input}} \leq I_\mathrm{E}^\mathrm{max} \tag{3}$$

$$i_\mathrm{W} = \frac{\sum_{sc \in Sc} m_{sc,\mathrm{W}}^\mathrm{dil}}{M_\mathrm{W}^\mathrm{input}} \leq I_\mathrm{W}^\mathrm{max} \tag{4}$$

$$\sum_{i \in V^+, j \in V^-, \lambda \in \Lambda} t_{i,j}^\lambda \leq 2N_\mathrm{sc} + 1 + N_\mathrm{split}^\mathrm{max} \tag{5}$$

$$m_{j,k}^\mathrm{feed} = \sum_{i \in V^+, \lambda \in \Lambda} m_{i,j,k}^\lambda \qquad \forall j \in V^-, k \in \{F, S\} \tag{6}$$

$$m_{j,\mathrm{W}}^\mathrm{feed} = \sum_{i \in V^+, \lambda \in \Lambda} m_{i,j,\mathrm{W}}^\lambda + m_{j,\mathrm{W}}^\mathrm{dil} \qquad \forall j \in V^- \tag{7}$$

$$m_\mathrm{a,W}^\mathrm{dil} = m_\mathrm{r,W}^\mathrm{dil} = 0 \tag{8}$$

$$m_{sc,k}^\mathrm{feed} = \sum_{j \in V^-, \lambda \in \Lambda} m_{sc,j,k}^\lambda \qquad \forall sc \in Sc, k \in K \tag{9}$$

$$0 \leq m_{i,j,k}^\lambda \leq M_k^\mathrm{max} \cdot t_{i,j}^\lambda \qquad \forall i \in V^+, j \in V^-, k \in K \tag{10}$$

$$\sum_{j \in V^-} m_{i,j,\mathrm{W}}^\mathrm{reject} = m_{s,\mathrm{W}}^\mathrm{feed} \cdot r_i \qquad \forall sc \in Sc \tag{11}$$

$$\sum_{j \in V^-} m_{sc,j,k}^\mathrm{reject} = m_{sc,k}^\mathrm{feed} \, r_{sc}^{p_{sc,k}} \qquad \forall sc \in Sc, k \in \{F, S\} \tag{12}$$

$$p_{sc,k} - P_{k,d}^\mathrm{pre} \leq (1 - b_{sc,d}) \qquad \forall sc \in Sc, d \in D, k \in \{F, S\} \tag{13}$$

$$p_{sc,k} - P_{k,d}^\mathrm{pre} \geq -(1 - b_{sc,d}) \qquad \forall sc \in Sc, d \in D, k \in \{F, S\} \tag{14}$$

$$\sum_{d \in D} b_{sc,d} = 1 \qquad \forall sc \in Sc \tag{15}$$

$$\sum_{j \in V^-} m^{\text{reject}}_{sc,j,F} \leq C^{\max} \sum_{j \in V^-} m^{\text{reject}}_{sc,j,\text{W}} \qquad \forall sc \in Sc \tag{16}$$

$$\sum_{sc \in Sc} m^{\text{accept}}_{\text{in},sc,k} = M^{\text{input}}_{k} \qquad \forall k \in K \tag{17}$$

$$\sum_{j \in V^-} t^{\text{reject}}_{\text{in},j} = 0 \tag{18}$$

$$t^{\text{accept}}_{\text{in},sc_1} = 1 \tag{19}$$

$$\sum_{i \in V^+} t^{\text{accept}}_{i,\text{r}} = \sum_{i \in V^+} t^{\text{reject}}_{i,\text{a}} = 0 \tag{20}$$

$$m^{\lambda}_{i,j,k} = \sum_{l \in V^-} m^{\lambda}_{i,l,k} \frac{w^{\lambda}_{i,j}}{W^{\text{inkr}}} \qquad \forall i \in V^+, j \in V^-, \lambda \in \Lambda, k \in \{F, S\} \tag{21}$$

$$0 \leq \frac{w^{\lambda}_{i,j}}{W^{\text{inkr}}} \leq t^{\lambda}_{i,j} \qquad \forall i \in V^+, j \in V^-, \lambda \in \Lambda \tag{22}$$

$$\sum_{\forall j \in V^-} \frac{w^{\lambda}_{i,j}}{W^{\text{inkr}}} = 1 \qquad \forall i \in V^+, \lambda \in \Lambda \tag{23}$$

Both passage ratios depend on the screen design, e.g., on its slot width or rotor geometry. Based on experimental data of [1], we derived passage ratios of different screen designs and fitted an exponential function to this data using least square approximation, cf. Fig. 2. The selection of a design out of a set of possible designs with passage ratios derived in preprocessing is modeled in (13) to (15). (16) limits the consistency to avoid a blinding of the basket. The sum of mass flows of all connections originating from the input node equals the total input, cf. (17), reject connections originating from the input are excluded in (18). (19) ensures one connection from the input to the first screen. We exclude unreasonable connections in (20), forbidding reject connections to the system accept as well as accept connections to the system reject. Splitting a screen's outgoing flow is possible. The

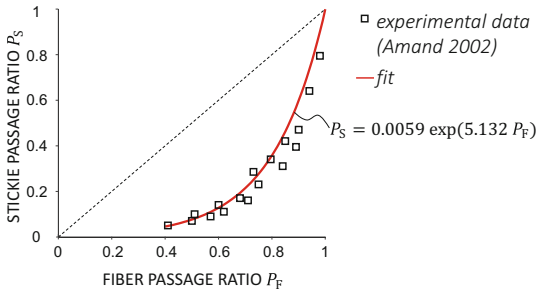


Fig. 2 Measured passage ratios of different pressure screens [1] and fitted relation

integer variable $w^{\lambda}_{i,j}$ divided by W^{inkr} denotes the partition of the total output flow of node $i \in V^+$ through the accept or reject connection (i, j) with $j \in V^-$ and $\lambda \in \Lambda$, cf. (21). Discretizing $w^{\lambda}_{i,j}$ speeds up the optimization. (22) and (23) ensure that we can only split between existing connections and that the splitting ratios sum up to 100%.

4 Computational Results

We use MATLAB 2017a with the toolboxes YALMIP [5] and OPTI [2] to implement the problem, and SCIP 5.0.1 [4] to solve it to proven global optimality. The sticky input flow and the volume flow are normalized to $M_{\text{S}}^{\text{input}} = M_{\text{W}}^{\text{input}} = 100$. An input consistency of $C^{\text{input}} = M_{\text{F}}^{\text{input}} / M_{\text{W}}^{\text{input}} = 1.5\%$ is set. The pipe capacity is limited to $M_{\text{W}}^{\max} = 4\, M_{\text{W}}^{\text{input}} (1 + I_{\text{W}}^{\max})$, $M_{\text{F}}^{\max} = C^{\max}\, M_{\text{W}}^{\max}$ and $M_{\text{S}}^{\max} = 8\, M_{\text{S}}^{\text{input}}$. i_W is limited to $I_{\text{W}}^{\max} = 20\%$ and i_E is unlimited. For the results in Fig 3, left, ten and for the results right five screen designs are considered. Figure 3, left, shows the trade-off between sticky reduction, a measure for the quality, and fiber yield. Depending on the weight of the objectives, three regions with different optimal layouts can be identified. By including the design decision for each screen, the system's performance can be increased. The influence of additional connections and

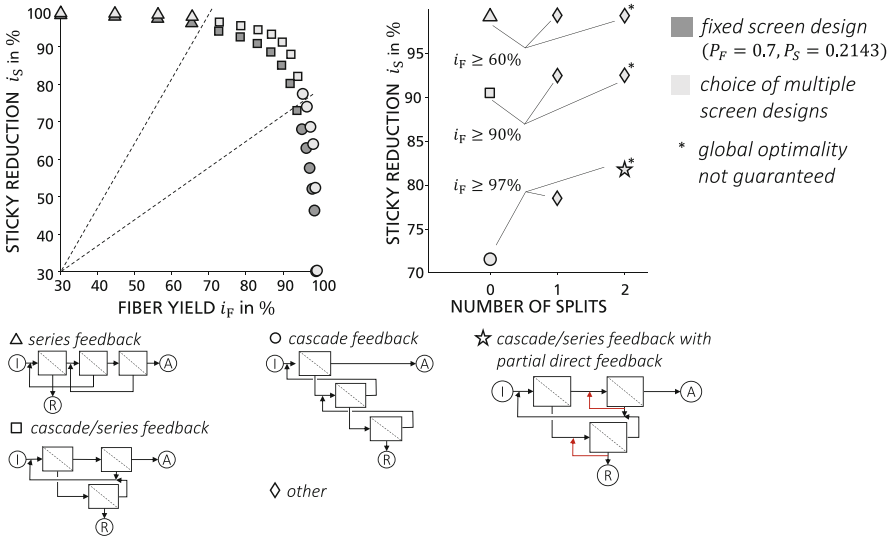


Fig. 3 *Left:* Trade-off between sticky reduction and fiber yield for optimized layout and operation with and without optimization of the screens' design. *Right:* Performance increase due to additional connections *Below:* Optimal layouts

a possible splitting of the streams is shown in Fig. 3, right. If a high fiber yield i_F is desired, splitting offers significant benefits.

5 Conclusion

We extended the MINLP introduced in [3] and enlarged the scope of the optimization. Our results showed room for improving fine screening systems by using optimization techniques, especially if the system layout and the design of single pressure screens are optimized simultaneously.

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. Amand, F.: Optimization of stickies removal in screens and cleaners. In: Recent Advances in Paper Recycling – Stickies, pp. 78–125. Wisconsin (2002)
2. Currie, J., Wilson, D.I.: OPTI: lowering the barrier between open source optimizers and the industrial MATLAB user. In: Foundations of Computer-Aided Process Operations. Georgia (2012)
3. Fügenschuh, A., Hayn, C., Michaels, D.: Mixed-integer linear methods for layout-optimization of screening systems in recovered paper production. Optim. Eng. **15**(2), 533–573 (2014)
4. Gleixner, A., et al.: The SCIP Optimization Suite 5.0. Zuse Institute, Berlin (2017)
5. Löfberg, J.: YALMIP: a toolbox for modeling and optimization in MATLAB. In: Proceedings of the CACSD Conference. IEEE, Piscataway (2004)
6. Steenberg, B.: Principles of screening system design: Studies in screening theory I. Sven. Papperst. **56**(20), 771–778 (1953)
7. Verband Deutscher Papierfabriken e.V.: Papier 2017. Bonn (2017)

Computation of Stable Honeycomb Structures for Additive Manufacturing

Martin Bähr, Georg Radow, Michael Breuß, and Armin Fügenschuh

1 Introduction

In certain additive manufacturing (AM) processes of industrial interest, the task arises to build up structures layer-wise from bottom to top in a purely vertical manner. Such orthographic structures are of specific interest if the printed material is soft at the moment when it is applied, as for instance in wire-arc AM where molten wire is transferred in droplets during printing. However, the question arises how to construct in a convenient way a printed orthographic structure that is structurally as stable as possible. In this paper, we proceed in two stages towards that goal.

In the first stage, we consider the automatic construction of a honeycomb structure through centroidal Voronoi tessellations (CVTs), given the boundary shape of a structure. In doing this we employ Lloyd's algorithm in two different realisations. For computing the incorporated Voronoi tessellation (VT), which is the crucial point when implementing this method, we consider either the use of a geometric optimisation method based on a Delaunay triangulation (DT) or the Eikonal equation which is a hyperbolic partial differential equation (PDE).

While finding a DT as the dual graph of the Voronoi diagram is based on geometric arguments, the Eikonal based approach makes use of a discretisation of the corresponding PDE. Thereby the number of generators used for constructing the VT is a design parameter. This is an important, beneficial aspect in practice, since it yields a way to take into account for instance total weight of a planned structure in the design process.

In another stage of the process, we consider the arising graph of the computed honeycomb structure as input for a specific routing scheme, cf. [6] for details. The

M. Bähr · G. Radow (✉) · M. Breuß · A. Fügenschuh
Brandenburg University of Technology, Cottbus, Germany
e-mail: martin.baehr@b-tu.de; radow@b-tu.de; breuss@b-tu.de; fuegenschuh@b-tu.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_45

routing for a single layer aims to (1) maximise the time before revisiting the same point during printing, and (2) minimise interrupts inside the printing process. While the first aspect allows for a maximal possible hardening time, the second aims to finish the process as fast as possible. Another important issue of the routing scheme is traversing the structure in an optimal way when applying the material layer by layer, which serves to enhance the stability of the printed structure.

2 Problem Formulation

Our region of interest is a simply connected compact set $\Omega \subset \mathbb{R}^2$, which represents one layer of the planned work piece in the printing process. Especially for cost reduction we aim to find a subset of Ω that is to be printed. This subset should include the boundary $\partial\Omega$ so that the exterior shape of the printed work piece stays the same. A trivial solution might be to only print the boundary $\partial\Omega$, however the resulting work piece may not be usable for the intended purpose due to stability issues.

Instead of the trivial solution we aim to find a planar graph where the edges represent the printed segments. This can be realised through a VT, which is a pair of points $X = (x_1, \ldots, x_m)$ with $x_i \in \Omega$ and a partition $\{V_i\}_{i=1}^m$ with $V_i \subset \Omega$, where

$$V_i = \left\{x \in \Omega : \|x - x_i\| < \|x - x_j\|,\, j \in \{1, \ldots, m\} \setminus \{i\}\right\} \tag{1}$$

where $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^2$. One can see $V_i \cap V_j = \emptyset$ for all $i \neq j$ and $\bigcup_{i=1}^m \overline{V}_i = \Omega$. The printed segments then consist of $\bigcup_{i=1}^m \partial V_i$, i.e., the border between two Voronoi cells V_i, V_j and the boundary $\partial\Omega$.

Since the Voronoi cells depend on the generators X, we also write $\mathscr{V}(X)$ for the set of all Voronoi cells and with $V(x_i) \in \mathscr{V}(X)$ we refer to the cell generated by x_i. If in each cell the generator coincides with the centre of mass, a VT is called a CVT. To compute the centre of mass, a density function or stress map $\rho : \Omega \to [0, \infty)$ is introduced. This stress map may be chosen to enhance stability in certain regions [10]. Formally the generators X^* of a CVT can be characterised through the minimisation of an energy functional [4]:

$$\mathscr{E}(X) = \sum_{i=1}^{m} \int\limits_{V(x_i)} \rho(x)\|x - x_i\|^2 \mathrm{d}x. \tag{2}$$

One of the basic methods for finding generators X^* of a CVT is Lloyd's algorithm [9], which is a fixed point iteration consisting of alternatingly computing the Voronoi cells $\mathscr{V}(X)$ and replacing the generator x_i with the centre of mass in V_i, i.e.,

$$x_i \leftarrow \frac{\int\limits_{V(x_i)} x\rho(x)\mathrm{d}x}{\int\limits_{V(x_i)} \rho(x)\mathrm{d}x}, \qquad \forall i \in \{1, \ldots, m\}. \tag{3}$$

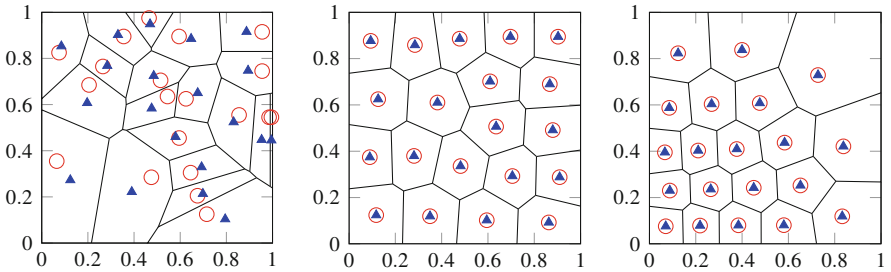


Fig. 1 Computed VTs by using the geometric approach for $m = 20$ generators (red circles) and centroids (blue triangles). *(Left)* Randomly distributed generators, *(middle)* CVT with trivial density $\rho = 1$, *(right)* CVT where the density is a Gaussian centred at the origin

Other methods for finding CVTs can be found, e.g. in [3, 7, 8] and the references therein. Increasing the number of generators leads to the hexagonal honeycomb form of the Voronoi cells, cf. [2], which is a structure of high strength-to-weight ratio.

The differences between VT as well as CVT for different ρ are shown in Fig. 1. All three experiments are computed with the geometric approach discussed in Sect. 3.1.

3 Computing Voronoi Tessellations

In this paper we want to compare two methods for finding VTs, the geometric or graph based approach utilising a DT, and a PDE-based approach utilising fast marching (FM).

3.1 Voronoi Tessellations Through Delaunay Triangulations

Voronoi tessellations can be found as the dual graph of a DT. This can be regarded as the typical approach for finding VTs [7, 12].

Indeed, if one finds a DT for the generators X, then the circumcentres of the triangles coincide with nodes of the graph, that consist of the boundary lines between the Voronoi cells. With this approach, special care must be taken for finding the nodes on $\partial\Omega$.

There are multiple resources available for finding VTs. One of these is the Computational Geometry Algorithms Library (CGAL) [5]. Also in MATLAB one can find VTs directly with the command `voronoin` based on the Qhull algorithm [1].

In our numerical experiments based on the geometric approach we use MATLAB. To get the boundary points on $\partial\Omega$ with this method, we mirror those generators on $\partial\Omega$, which are 'close enough' to this boundary. If Ω is a polygon, this means to find those generators x_i, which are the closest generators to some part of the boundary $\partial\Omega$, and mirror them on the corresponding line segment, since some part of the line segment will be a part of $V(x_i)$.

For the realisation of Lloyd's algorithm with the discussed geometric approach, the centres of mass are computed in an approximative manner. For (3) the density ρ is only sampled in the centre of a triangle between a generator x_i and two consecutive nodes on the border of its Voronoi cell $V(x_i)$.

3.2 PDE-Based Approach

The computation of a VT involves the Euclidean distance function. It is well-known, that for a convex domain Ω the geodesic distance function d, defined as the shortest path contained in Ω between two points, is equivalent to the Euclidean distance function. Due to this fact, any method computing geodesic distances can be used to generate a VT. Finding the shortest paths on meshes is often called discrete geodesic problem. One possibility to compute the discrete geodesic is solving a PDE on the underlying mesh.

One common approach to geodesic distance computation is solving the nonlinear Eikonal equation

$$\|\nabla d(x)\| = 1, \quad x \in \Omega \setminus \Omega_0 \tag{4}$$

with the boundary condition

$$d(s) = 0, \quad s \in \Omega_0 \tag{5}$$

where Ω_0 is a subset of Ω, see also [2]. The underlying PDE represents the shortest arrival time of a wavefront from the initial point s to every point x in the computational domain, whereby the wavefront moves in its normal direction with constant unit speed. A solution of (4)–(5) can be computed efficiently by the FM method proposed by Sethian [11].

The basic FM algorithm on an orthogonal grid makes use of the standard first-order upwind scheme for the numerical approximation of (4). Subsequently, a specific causality relationship, where the information of arrival times is propagated "downwind", is applied. Furthermore, FM can be applied on non-uniform grids and triangulated meshes and additionally high-order upwind schemes are also applicable, cf. [11]. The benefit of FM is its relatively low complexity of $\mathcal{O}(n \log n)$ by using a heap sort algorithm, where n is the number of points in the computational domain.

Computing a VT by using the FM method is done in the following manner: set $s = x_i$, $\Omega_0 = X$ and start simultaneously a wavefront from each of the generators x_i and thus generate distance maps (Voronoi cells). At the points where two wavefronts collide the border between two Voronoi cells is obtained.

Let $x^{(j)}$ denote the location on the grid and $\rho^{(j)}$ be the corresponding density value. Then the discretisation of (3) is

$$x_i \leftarrow \frac{\sum\limits_{j \in J} \rho^{(j)} x^{(j)}}{\sum\limits_{j \in J} \rho^{(j)}}, \quad J = \left\{ j : x^{(j)} \in V(x_i) \right\}, \qquad \forall i \in \{1, \ldots, m\}. \tag{6}$$

3.3 Comparison of Geometric and PDE-Based Approach

Let us now discuss some important properties of the two presented methods for computing a CVT. The geometric approach generates the exact VT, whereas the PDE-based method can only deliver an approximation.

In both methods the centres of mass x_i for constructing a CVT are computed by numerical approximation. For this reason, a significant factor in terms of the accuracy of x_i is the sampling rate of the underlying density. On the one hand, the sampling points of the geometric approach depend on the shape of the Voronoi cells, which is changing during the method. In contrast, for the PDE-based approach the sampling points consist of the grid points for all iterations.

With the geometric approach the threshold for the stopping criteria used for the Lloyd's algorithm can be set to arbitrarily small values. When using FM the generators can only move on a discrete grid, therefore a natural stopping criteria is reached, if the generators of two consecutive iterations are identical.

Finally let us mention, in AM often non-convex shapes are of interest. With the geometric approach this case requires more advanced implementations, see e.g. [12]. The PDE-based approach for non-convex shapes is significantly easier to handle.

4 Trajectory Optimisation

After the CVT is obtained the task of planning the printing process remains. The problem of finding an optimal trajectory is formulated as a mixed-integer linear program, which uses a min-cost flow on a time-expanded graph with further constraints that model the dissipation of heat in the nodes. Since in most cases the workpiece cannot be traversed in a continuous loop without stopping and restarting the laser welding process, the objective is to find a trajectory with a minimum amount of interrupts that additionally minimises the heat that is accumulated in each

node during production time. In particular, welding at the same node twice within a short amount of time should be avoided if possible. Details of this approach can be found in [6].

5 Results and Future Work

Let us give a qualitative example for the presented approach when considering a certain workpiece. The main stages are depicted in Fig. 2. There the Voronoi cells are computed with FM, subsequently the edges of the CVT are generated in a postprocessing step. Finally, the optimal trajectory is obtained with the routing scheme described in Sect. 4. This example leads to a model instance with 5397 variables and 2009 constraints, which is solved by IBM ILOG CPLEX 12.8.0 (default settings) on a MacBookPro with an Intel Core i7 running 8 parallel threads at 3.1 GHz clockspeed and 16 GB RAM in 2679 s to proven global optimality.

The optimal trajectory in Fig. 2 contains four deadhead transitions. In order to avoid these transitions, a graph with less nodes of uneven degree is preferable [6]. This may be achieved e.g., by adapting edges and nodes of the found CVT. We aim to address this in our future work.

Also we want to evaluate the two discussed methods for finding a CVT in a more quantitative manner. For this the extension of our approaches to non-convex three dimensional objects will be of importance. Through this evaluation we aim to find the most suitable method in the context of AM. Apart from FM also other PDE-based approaches for computing distances may be of interest.

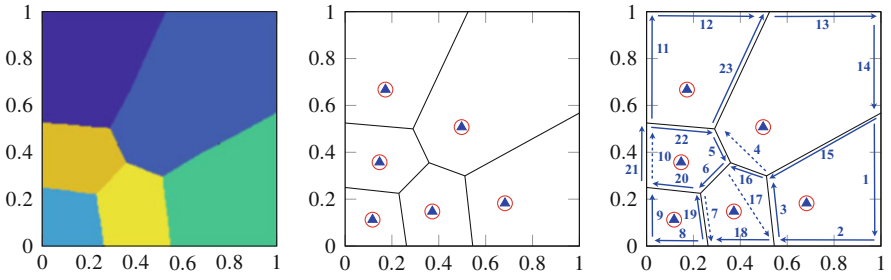


Fig. 2 Computed CVT by using FM for $m = 6$ randomly initialised generators, where the density is a Gaussian centred at the origin. *(Left)* Discrete CVT, where different colours corresponds to different Voronoi cells. Black colour points belong to multiple Voronoi cells. *(Middle)* The extracted graph from the discrete CVT. *(Right)* The trajectory of the welding torch. Dashed lines are deadhead transitions, straight lines are welded segments

References

1. Barber, C.B., Dobkin, D.P., Huhdanpaa, H.: The quickhull algorithm for convex hulls. ACM Trans. Math. Softw. **22**(4), 469–483 (1996)
2. Bronstein, A.M., Bronstein, M.M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer, New York (2009)
3. Du, Q., Emelianenko, M.: Acceleration schemes for computing centroidal Voronoi tessellations. Numer. Linear Algebra Appl. **13**, 173–192 (2006)
4. Du, Q., Faber, V., Gunzburger, M.: Centroidal Voronoi tessellations: applications and algorithms. SIAM Rev. **41**(4), 637–676 (1999)
5. Fabri, A., Giezeman, G.J., Kettner, L., Schirra, S., Schönherr, S.: On the design of CGAL a computational geometry algorithms library. Softw. Pract. Experience **30**(11), 1167–1202 (2000)
6. Fügenschuh, A., Bambach, M., Buhl, J.: Trajectory optimization for wire-arc additive manufacturing. In: 2018 International Conference on Operations Research (2018). Accepted for publication
7. Hateley, J.C., Wei, H., Chen, L.: Fast methods for computing centroidal Voronoi tessellations. J. Sci. Comput. **63**, 185–212 (2015)
8. Liu, Y., Wang, W., Lévy, B., Sun, F., Yan, D.M., Lu, L., Yang, C.: On centroidal Voronoi tessellation—energy smoothness and fast computation. ACM Trans. Graph. **28**(4), 1–17 (2009)
9. Lloyd, S.: Least squares quantization in PCM. IEEE Trans. Inf. Theory **28**(2), 129–137 (1982)
10. Lu, L., Sharf, A., Zhao, H., Wei, Y., Fan, Q., Chen, X., Savoye, Y., Tu, C., Cohen-Or, D., Chen, B.: Build-to-last: strength to weight 3D printed objects. ACM Trans. Graph. **33**(4), 97 (2014)
11. Sethian, J.A.: Level Set Methods and Fast Marching Methods, 2nd edn. Cambridge Press, Cambridge (1996)
12. Tournois, J., Alliez, P., Devillers, O.: 2D centroidal Voronoi tessellations with constraints. Numer. Math. Theory Methods Appl. **3**(2), 212–222 (2010)

Design and Optimization for Additive Manufacturing of Cellular Structures Using Linear Optimization

Christian Reintjes, Michael Hartisch, and Ulf Lorenz

1 Introduction and Motivation

The key benefit of Additive Manufacturing (AM), in areas of application such as lightweight construction, is that there are fewer manufacturing constrains than in classical manufacturing [1]. It is possible to implement a higher level of individualization and complexity [2]. In order to take advantage and fully exploit the resulting potential of these new manufacturing methods, it is important that the components are constructed and optimized by well adapted optimization methods in advance. We focus on optimization and additive manufacturing of cellular structures.

As a first step we used MIPs to optimize two-dimensional assembly spaces [3]. As a second step we present a MIP to describe a static load shall for a three-dimensional assembly space for the use case AM. It is shown in [4] that a two-dimensional robust truss topology design problem with beam elements is describable by a nonlinear semidefinite program considering nonlinearity on the cross-sectional areas of the beams and uncertainty in load-carrying. In contrast, we focus on large 3d-instances characterized by an appropriate assembly space and level of granularity, by neglecting nonlinear material behaviour. The target of the optimization is to reduce cost as well as material and improve the field of lightweight construction, with the objective of developing permissible and optimized constructions.

Furthermore, the results of the optimization process are integrated in a design tool embedded in a CAD system (Autodesk Inventor 2017 Pro). Hence, we are able to verify nonlinear material behaviour with a standard finite element analysis (FEA) subsequent to the optimization. The generated CAD drawing is convertible

C. Reintjes (✉) · M. Hartisch · U. Lorenz
Institute of Technology Management, University of Siegen, Siegen, Germany
e-mail: christian.reintjes@uni-siegen.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_46

to the stereolithography file format (STL) by means of a data interface. In a further step the STL file can be converted to numerical control data for the steering of the printing process and, as a result, finalise the optimization process by manufacturing an optimized component with minimal material consumption and cost.

2 Problem Setting and Modelling Approach

The considered assembly space is a three-dimensional cuboid modeled as a graph with edges—representing possible beam elements—and vertices—representing possible connection nodes. The dimensioning of the assembly space is defined by the number of connection nodes (X, Y, Z) in each direction in space (x, y, z). In Fig. 1 (left) the complete assembly space for $X = Y = Z = 5$ is given.

Each centre of a node acts as a force application point with at most 26 mounted beam elements, resulting from the arrangement of beam elements with an angular distance of 45° with respect to each principal axis direction. As an example a node with its possible beam elements is illustrated on the right hand side of Fig. 1. At each node an external vertical force can be transmitted into the assembly space, i.e., a static vertical load distribution is considered. Further, nodes can be specified as bearings exerting bearing reaction forces. The task is to assign beam elements with different cross section diameter (and thus different capacities regarding the permissible stress) to the edges such that the resulting structure is stable—in compliance with the laws of statics. The cross section diameter is constant along the entire length for a chosen beam element between two nodes.

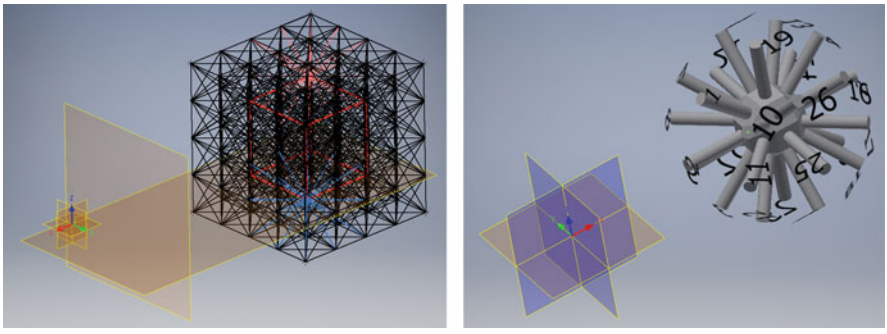


Fig. 1 Illustrative presentation of a simplified assembly space ($X = Y = Z = 5$) with a highlighted particular case and the associated node (assembly element simplified) with the maximum number of beam elements (cf. green-labeled area and red marked beam elements)

3 Model Formulation

The objective function aims at minimizing the cost of the used beam elements which are needed to create a statically determined cellular structure under the influence of external forces exerted perpendicularly. Restrictions (1) to (3) demand a force equilibrium at every connecting node (reference point) for the external forces Q_i and active forces $F_{i,j}$ of the beam elements in all spatial directions in space. The decomposition of the forces in x, y, z direction is obtained by applying the appropriate trigonometric function to the angle of revolution in relation to the principal axis (see e.g., $r_{i,j,x}$). The external forces Q_i are transmitted as force components in the respective equilibrium conditions.

$$\min \sum_{i \in V} \sum_{j \in V} \sum_{t \in T} B_{t,i,j} \cdot cost_t$$

$$\text{s.t.} \sum_{j \in NB_x(i)} F_{i,j} \cdot r_{i,j,x} + Q_{i,x} = 0 \qquad \forall i \in V \qquad (1)$$

$$\sum_{j \in NB_y(i)} F_{i,j} \cdot r_{i,j,y} + Q_{i,y} = 0 \qquad \forall i \in V \qquad (2)$$

$$\sum_{j \in NB_z(i)} F_{i,j} \cdot r_{i,j,z} + Q_{i,z} + R_{i,z} = 0 \qquad \forall i \in V \qquad (3)$$

$$\sum_{\substack{i \in V \\ Q_i \neq 0}} Q_{i,z} \cdot (L_{i,y} - L_{j,y}) + \sum_{k \in B} R_{k,z} \cdot (L_{k,y} - L_{j,y}) = 0 \qquad \forall j \in B \qquad (4)$$

$$\sum_{\substack{i \in V \\ Q_i \neq 0}} Q_{i,z} \cdot (L_{i,x} - L_{j,x}) + \sum_{k \in B} R_{k,z} \cdot (L_{k,x} - L_{j,x}) = 0 \qquad \forall j \in B \qquad (5)$$

$$L_{i,x} = (i \mod (xz - 1)) \mod x \qquad \forall i \in V \qquad (6)$$

$$L_{i,y} = \left\lfloor \frac{i-1}{xz} \right\rfloor \qquad \forall i \in V \qquad (7)$$

$$F_{i,j} \leq M \cdot x_{i,j} \qquad \forall i, j \in V \qquad (8)$$

$$F_{i,j} = -F_{j,i} \qquad \forall i, j \in V \qquad (9)$$

$$F_{i,j} \leq \sum_{t \in T} c_t \cdot B_{t,i,j} \qquad \forall i, j \in V \qquad (10)$$

$$B_{t,i,j} = B_{t,j,i} \qquad \forall i, j \in V, t \in T \qquad (11)$$

$$\sum_{t \in T} B_{t,i,j} = x_{i,j} \qquad \forall i, j \in V \qquad (12)$$

$$x_{i,j} = x_{j,i} \qquad \forall i, j \in V \qquad (13)$$

$$R_{i,z} = 0 \qquad \forall i \in V \setminus B \qquad (14)$$

$$x_{i,j}, B_{t,i,j} \in \{0, 1\} \qquad \forall i, j \in V, t \in T \qquad (15)$$

Constraint (4) and (5) define the statical moment of an area which results from the external forces and the bearing reaction forces $R_{k,z}$. Analogous to the restrictions (1) to (3) all three spatial directions in space are considered, whereat one is obsolete since only pure vertical forces are exerted. Constraints (3) (see $R_{z,i}$), (4) and (5) unify the externally and internally statically determinacy. Constraints (6) and (7) are of use for the necessary lever arms and define the specific positions $L_{i,x}$, $L_{i,y}$ of a node i in the assembly space: $L_{i,x}$, $L_{i,y}$ are defined as the numbers of the levels—with regard to the respective direction—measured from the origin of the coordinate system ($L_{1,x} = L_{1,y} = 1$) (Table 1).

Table 1 Decision variables, parameters and sets

Symbol	Definition
$B_{t,i,j} \in \{0, 1\}$	Binary variable indicating whether bar of type $t \in T$ is present between $i \in V$ and $j \in V$
$F_{i,j} \in \mathbb{R}$	Flow of forces between nodes $i \in V$ and $j \in V$
$x_{i,j} \in \{0, 1\}$	Binary variable indicating whether a bar is present between nodes $i \in V$ and $j \in V$
$L_{i,x}, L_{i,y}$	Level number measured from the reference node $i = 1$ in the direction of space x or y
$R_{i,z}$	Bearing reaction force in z direction
$X, Y, Z \in \mathbb{N}$	Length, width and height of the assembly space stated as the number of connection nodes
$r_{i,j,x}, r_{i,j,y}, r_{i,j,z} \in \{0, \frac{\sqrt{2}}{2}, 1\}$	Force component at node i relative to the reference plane x, y, z, caused by beam structure between $i \in V$ and $j \in V$
$V = \{1, \ldots, XYZ\}$	Set of connecting nodes (possible truss joints)
$T = \{0, 1, \ldots, s\}$	Set of different beam types
$c_t \in \mathbb{R}_+$	Capacity of beam type t or allowable normal stress $\sigma_N = \sigma_{allow}$ of a certain beam type t
$M \in \mathbb{R}$	Big M—maximum capacity of the most robust beam type
$cost_t \in \mathbb{R}$	Cost of beam type t
$Q_{i,x}, Q_{i,y}, Q_{i,z} \in \mathbb{R}_+$	Force component in x, y or z direction of the applied concentrated load at node $i \in V$
$NB(i) \subseteq V$	Set of neighboring nodes of i
$NB_x(i), NB_y(i), NB_z(i) \subseteq NB(i)$	Set of neighboring nodes of i which have a force component in x, y or z direction in space
$B \subseteq V$	Set of nodes acting as bearings

Constraint (8) ensures that only used beams can transfer forces. Constraint (9) represents Newton's third law and constraint (10) limits the force in a beam with regard to the permissible force of the used beam type. Constraint (11) guarantees that a specific beam type is selected if a beam is used. The two last constrains demand a used beam to go both ways (13). Constraint (14) sets the bearing reaction forces of non-bearing nodes to zero.

4 Computational Results, CAD Integration and FEA

The two instances introduced in the following consider a positioning of a static area load (see Fig. 2, left).[1] The computation time (manually interrupted) was 10 h 51 min. for the first instance and 3 h 50 min. for the second. 12 resp. 9 permissible solutions were determined, whereby the duality gap was 55.73% resp. 6651%. The assembly space consists of a width, height and depth of 120 mm. These dimensions results from a number of nodes of $X = Y = Z = 6$ in all spatial directions, at a standardized beam element of 20 mm. Hence, there are 216 connecting nodes. The four corner points of the first plane in z-direction are defined as bearings. It is predetermined that the top level in z-direction is fully developed with beam elements. The beam element diameters 2, 4, 6 and 8 mm and in addition 10 mm for the second instance together with the associated $c_t \in \mathbb{R}_+$ and $cost_t \in \mathbb{R}$ are passed as parameters. Furthermore, the nonlinear material behavior of the construction proposal was investigated by means of an FEA analysis. Figure 2 (right) shows the boundary conditions and the deformation behavior (total deformation) as a fringe plot.

The second instance is identical to instance one with respect to the dimensions. A targeted control of the structure's density was enabled through customization of the optimization preferences. The exploitation of a beam element with a cross section (diameter) of 4 mm is penalized with an increase of 50% in material cost. Furthermore, a beam element with a diameter of 10 mm is passed as a parameter. The usage of a 4 mm beam elements is reduced to zero (see Table 2), whereby the number of used beam elements is reduced by 4.22%, resulting in a less dense cellular structure.

[1] The calculation were executed on a workstation with an Intel Xeon E5-2637 v3 (3.5 GHz) and 128 GB RAM using CPLEX Version 12.6.1. The CAD construction (Inventor 2017 Pro) as well as the FEM simulation (Ansys 18.2) was performed on a workstation with an Intel Core i7-4710HQ (2.5 GHz), NVIDIA GeForce GTX 860M (4096 MB) and 16 GB RAM.

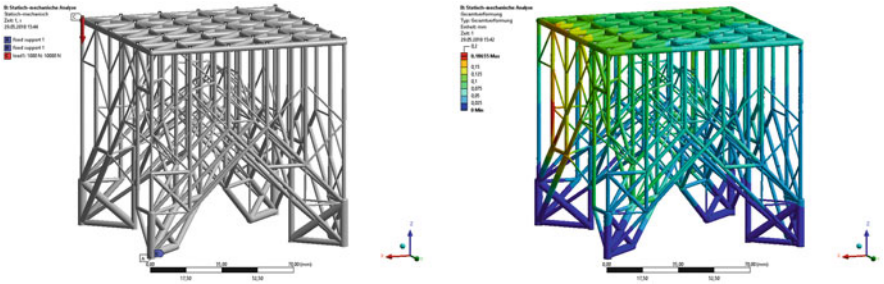


Fig. 2 Construction proposal based on linear optimization (left) and the result of the associated FEA (right)

Table 2 Adaptation of the density of the installed truss elements

Diameter	Instance 1 [3336 possible bars]	Instance 2 [3336 possible bars]
2 mm	200	102
4 mm	**378**	**0**
6 mm	69	407
8 mm	18	8
10 mm	–	7
$\sum$	**665**	**524**
Reduction of	80.07%	84.29%

5 Conclusion and Outlook

Our presented MIP demonstrates that the usage of mathematical programming in the AM domain can be used to realize constructions with minimal material consumption and costs. The optimization model is used for a construction made up of cellular structures resulting from a static load shall. A control of the density of the installed beam elements is possible through customization of the optimization preferences. The automated construction in a CAD system allows a numerically simulation of nonlinear material behaviour on the basis of the optimization results. The model is planned as a support tool for engineers.

References

1. Chu, C., Graf, G., Rosen, D.W.: Design for additive manufacturing of cellular structures. Comput. Aided Des. Appl. **5**(5), 686–696 (2008)
2. Brackett, D., Ashcroft, I., Hague, R.: Topology optimization for additive manufacturing. In: Proceedings of the Solid Freeform Fabrication Symposium, Austin, pp. 348–362 (2011)

3. Reintjes, C., Hartisch, M., Lorenz, U.: Lattice structure design with linear optimization for additive manufacturing as an initial design in the field of generative design. In: Proceedings 2017 of the Conference on Operations Research, pp. 451–457. Springer, Berlin (2017)
4. Gally, T., Gehb, C.M., Kolvenbach, P., Kuttich, A., Pfetsch, M.E., Ulbrich, S.: Robust truss topology design with beam elements via mixed integer nonlinear semidefinite programming. In: Uncertainty in Mechanical Engineering II, pp. 229–238. Trans Tech Publications (2015)

Machine Learning and Metaheuristics for Black-Box Optimization of Product Families: A Case-Study Investigating Solution Quality vs. Computational Overhead

David Stenger, Lena C. Altherr, and Dirk Abel

1 Introduction

Large production volumes and competitive markets require particularly cost-efficient product designs. Using common parts across product families can significantly reduce production costs, while maintaining product variety. In platform-based product families, products are derived from a common platform, yet each product variant satisfies different customer requirements. High commonality across the product family typically results in savings of variance dependent cost, e.g., for tooling whereas individual product costs such as material costs may rise due to over-dimensioning of single products. Thus, choosing the right amount of commonality is a key issue in product family design. To evaluate the technical feasibility of design proposals, typically a combination of tools from multi-domain virtual prototyping like Finite Element Method or Finite Volume Method are used. However, these tools often require substantial computational effort for just a single design evaluation. Therefore, mathematical optimization is a promising alternative to brute force or experience-based methods. Goal is to find optimal product families with as little expensive design simulation as possible at a reasonable overhead.

D. Stenger (✉) · D. Abel
Institute of Automatic Control (IRT), Faculty of Mechanical Engineering, RWTH Aachen University, Aachen, Germany
e-mail: D.Stenger@irt.rwth-aachen.de; D.Abel@irt.rwth-aachen.de

L. C. Altherr
Chair of Fluid Systems, Department of Mechanical Engineering, TU Darmstadt, Darmstadt, Germany
e-mail: lena.altherr@fst.tu-darmstadt.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_47

2 Problem Description

In our problem setup, the evaluation of the feasibility and individual cost of a design proposal for a single product are obtained through a black-box interface: We may query the respective values for a specific design, but we cannot obtain gradient information or use favourable characteristics (e.g., convexity) of the objective function or the constraints, because we cannot make any assumptions on their analytic form. Modelling different discrete design options requires continuous as well as integer decision variables. This leads in a natural way to a mixed-integer nonlinear black-box optimization problem. Moreover, the trade-off between savings due to increased commonality and expensive overdimensioning of single products has to be explored. A general cost model of production lines and logistics being not available in early design phases prevents resolving this by aggregation of costs. Instead, the conflict of goals between low platform variance in each platform parameter and low cumulated individual product costs is explored, yielding a constraint mixed-integer nonlinear multi-objective program:

$$\min_X [C^{\text{ind}}(X), \Lambda_1(X_{*,1}), \ldots, \Lambda_m(X_{*,m})]^T \tag{1}$$

$$\text{s.t.}: \quad \forall l \in L, \forall i \in I: \quad g_{l,i}(X_{i,*}) \leq g_{l,i}^{\max} \tag{2}$$

$$\forall j \in J, \forall i \in I: \quad a_j \leq X_{i,j} \leq b_j \tag{3}$$

$$X \in \mathbb{Z}^p \times \mathbb{R}^{n-p} \tag{4}$$

where

$X = (X_{i,j})$	product family with value $X_{i,j}$ for design parameter j of product i,
$C^{\text{ind}}(X)$:	individual product costs obtained through black-box interface,
$\Lambda_j(X_{*,j})$:	number of different values (platform variance) of parameter j,
L:	product attributes obtained through black-box interface,
I:	set of product variants of the family,
$g_{l,i}(X_{i,*})$:	black-box response for attribute l of product i,
$g_{l,i}^{\max}$:	product specific bounds on the black-box responses,
a_j, b_j:	lower and upper bound of design parameter j.

3 State of the Art

We deal with the so-called 'joint product platform selection and design' problem [1], in which an optimal platform configuration (stating which parameters are individual and which are shared among the products) and optimal designs of each product are found simultaneously. Numerous approaches of tackling this problem have been proposed. For a comprehensive review we refer to [2]. One example is a Genetic Algorithm (GA) with product family specific crossover and mutation operators [3].

However, in case of black-box optimization, metaheuristic approaches may suffer from a high number of necessary design simulations.

Metamodel-based optimization is a black-box optimization approach with which one tries to maximize the usage of the information gained with each design evaluation by constructing a 'fast-to-evaluate' data-driven surrogate model for the 'slow-to-evaluate' simulation model. Promising designs are chosen by optimization of an infill criterion on the metamodel, and are evaluated on the 'slow-to-evaluate' simulation model. Results of the expensive design evaluations are subsequently used to iteratively update the metamodel, thus improving its quality in promising regions. Recently, the general framework MISO for metamodel-based mixed-integer optimization has been developed [4], however only single-objective problems and no black-box constraints are covered. While the general framework SOCEMO covers multi-objective metamodel-based optimization [5], it still has to be extended to problems with black-box constraints and integer variables. To the best of the authors' knowledge, presently no general framework covering all aspects of the presented optimization problem exists.

In previous work [6] we introduced a metamodel-based optimization approach which relies on a Gaussian Process Regression (GPR) metamodel. It uses a bounded tree search to optimize statistical lower bounds as an infill criterion. In this work, we seek to improve this method by investigating different GPR models and use a GA to maximize the expected improvement. GPR is also known as Kriging [7]. We use the term GPR because some key ideas used in this work were developed in the machine learning community where the term GPR is used predominantly.

4 Metamodel-Based Optimization of Product Families

Using the concept depicted in Fig. 1, we apply metamodel-based optimization to the domain of product family optimization. After an initial sampling (a), GPR metamodels of all relevant single product responses are built and in the following iterations improved (b). GPR models are non-parametric probabilistic regression models used in supervised machine learning [8]. They can also be used to model dynamic processes which is an important aspect of control engineering. The predictive distribution of black-box response g at location $\hat{x}$ obtained by GPR is a normal distribution $\widetilde{g}(\hat{x}) \sim \mathcal{N}(\bar{g}(\hat{x}), \sigma_g(\hat{x}))$ with predicted mean $\bar{g}(\hat{x})$ and predicted standard deviation $\sigma_g(\hat{x})$.

At each iteration, we combine the growing archive of evaluated single product designs to the current solution set of product families (c) using a tree search algorithm introduced in [6]. We use the popular NSGA-II in combination with product family specific mutation and crossover operators introduced in [3] to search the response surface for candidate product families (d). The infill criterion 'expected improvement with constraints' (EIC) [9] is optimized by the GA.

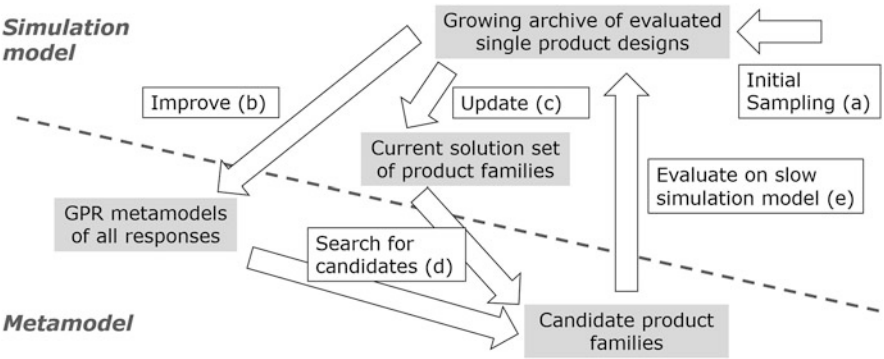


Fig. 1 Overview of the metamodel-based optimization approach

The EIC of a candidate product family $\hat{X} = (\hat{X}_{i,j})$ with platform variance $\hat{\Lambda}$ can be written as:

$$\mathrm{EIC}(\hat{X}) = \prod_{i=1}^{n} \prod_{l=1}^{k} \mathrm{P}(\widetilde{g}_{l,i}(\hat{X}_{i,*}) < g_{l,i}^{\max}) \mathrm{E}[\max\{0, C_{\min}^{\mathrm{ind}}(\hat{\Lambda}) - \widetilde{C}^{\mathrm{ind}}(\hat{X})\}] \tag{5}$$

The first term takes the probability of all product variants fulfilling their black-box constraints into account. The second term balances the exploitation of low predicted costs with the exploration of areas with uncertain cost information. $C_{\min}^{\mathrm{ind}}(\hat{\Lambda})$ denotes the cheapest product family found so far with platform variance $\hat{\Lambda}$. $\widetilde{C}^{\mathrm{ind}}(\hat{X})$ denotes the normal predictive distribution of individual costs obtained by GPR. Promising product families are then evaluated on the simulation model (e) and added to the archive of evaluated designs.

There are two major challenges when using GPR in the framework described above. First the metamodel needs to be retrained frequently as an increasing number of designs are simulated. Secondly, choosing a suitable mean function for the GPR is not straight forward and heavily influences the generalization capabilities of GPR. A *dense GPR model with constant mean function (Alg. I)*, also known as ordinary kriging, serves as a baseline.[1] Alg. I has a computational complexity of n^3, where n is the number of training points, which makes frequent refitting unsuitable. This issue is addressed by using a *Sparse GPR with constant mean (Alg. II)*. We use the FITC [10] approximation. The inducing points are chosen as a subset of the training data by maximization of the likelihood. In an attempt to increase the extrapolation capabilities of the model *Sparse GPR with linear mean (Alg. III)* is employed. In order to circumvent the manual trend function selection and to increase the

[1] In all cases, an anisotropic squared exponential kernel and a Gaussian likelihood function with a noise level fixed to 10^{-10} are used. Hyper parameters θ of the model are chosen by maximization of the log likelihood.

flexibility of the model, a *Bayesian model selection technique (blind kriging)* [11] is used in a novel combination with sparse GPR to choose meaningful trend functions automatically *(Alg. IV)*. Solely the inducing points of the FITC approximation are used for trend function selection to bound computational overhead. All GPR models are implemented using the GPML toolbox [12] partly in combination with the ooDACE toolbox [11]. The algorithm is implemented in *MATLAB 2017b*.

5 Results

The proposed approaches are benchmarked on a real-world product family of electrical drives provided by Robert Bosch GmbH. The product family consists of 8 different products. Each of them is defined by 6 design parameters of which two are considered as potential platform parameters. In addition to individual product costs, the responses of four product attributes are approximated by GPR models due to their appearance within black-box constraints. To compare the results with an optimal reference solution, the design space is discretized and simulated full factorial. This way, in total 3865 single product designs are evaluated successfully, resulting in a total of 1.2×10^{23} different possible product families. An optimal reference solution set is generated by combining all feasibly evaluated designs using a tree search algorithm proposed in previous work [6].

Figure 2 shows that all metamodel-based approach significantly reduce the number of expensive black-box evaluations needed to reach the optimal reference solution, and that all approaches outperform random sampling. The approaches with constant mean (Alg. I and II) perform worse than those with linear (Alg. III) and

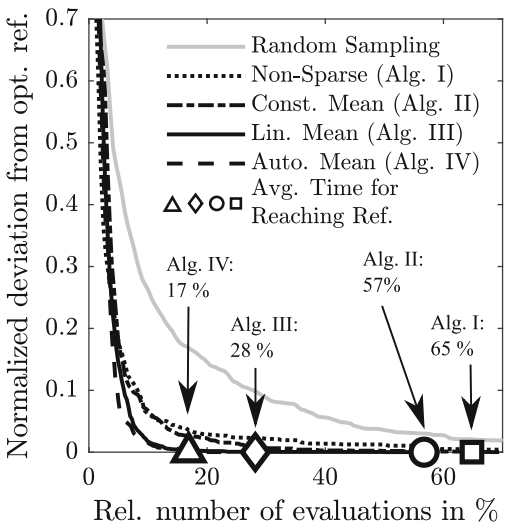


Fig. 2 Progress of Optimization. After the initial sampling the average normalized deviation from the optimal reference (opt. ref.) is 1

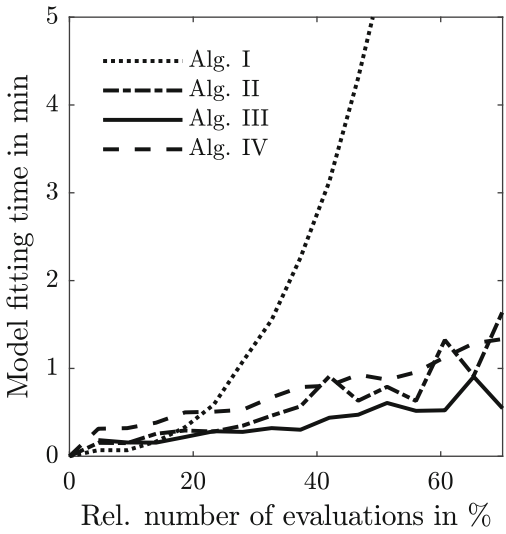


Fig. 3 Overhead for fitting the different GPR models for one black-box response

automatically chosen mean (Alg. IV). Alg. IV slightly outperforms Alg. III. The average overheads of Algs. I–IV are 8.5 h, 6.9 h, 1.6 h and 1.2 h, respectively. They are dominated by the time required for refitting the GPR Model. Figure 3 shows the fitting time for one of the five black-box responses. It increases with the data available, i.e., with the number of executed black-box evaluations. The fitting time of the non-sparse Alg. I shows approximately cubic growth. The increase in overhead due to trend-function selection of Alg. IV in comparison to Algs. II and III is not significant. Yet, with 5.3 s per design simulation, Alg. III has the lowest average overhead until reaching the reference solution.

6 Discussion and Outlook

It was shown that GPR in combination with a GA maximizing the expected improvement with constraints is able to efficiently optimize real world product families. The combination of automated trend function selection with sparse GPR allows for superior performance at reasonable computational overhead. We believe that the superior performance is caused by the improved generalization capability of the flexible trend function in comparison to constant or linear mean GPR. The Bayesian model selection technique ensures that only relevant basis functions are used. The slight increase in model fitting time is in our case compensated by better performance. If the goal is to minimize the overhead per design simulation, sparse GPR with linear mean should be used. We confirmed for our case that sparse GPR allows for similar predictive performance in comparison to normal GPR, while significantly reducing computational overhead. In conclusion, we used state of the

art machine learning algorithms in combination with a problem specific GA to enable engineers to systematically optimize complex product families. In future research, we plan to combine further product family specific heuristics with sparse GPR and compare their performance with the one of the GA.

References

1. Fujita, K., Yoshida, H.: Product variety optimization: simultaneous optimization of module combination and module attributes. In: Proceedings of DETC'01, 2001 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference (2001)
2. Simpson, T.W., et al. (eds.).: Advances in Product Family and Product Platform Design. Springer, New York (2014)
3. Khajavirad, A., et al.: An efficient decomposed multiobjective genetic algorithm for solving the joint product platform selection and product family design problem with generalized commonality. Struct. Multidiscip. Optim. **39**(2), 187–201 (2009)
4. Müller, J.: MISO: mixed-integer surrogate optimization framework. Optim. Eng. **17**(1), 177–203 (2016)
5. Müller, J.: SOCEMO: surrogate optimization of computationally expensive multiobjective problems. INFORMS J. Comput. **29**(4), 581–596 (2017)
6. Stenger, D., et al.: Product Family Design Optimization Using Model-Based Engineering Techniques. Proceedings of the Conference on Operations Research 2017, pp. 495–502. Springer, Cham (2018)
7. Krige, D.G.: A statistical approach to some basic mine valuation problems on the Witwatersrand. J. South. Afr. Inst. Min. Metall. **52**(6), 119–139 (1951)
8. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. MIT Press, Cambridge, MA (2006). ISBN 026218253X
9. Forrester, A.I., Keane, A.J.: Recent advances in surrogate-based optimization. Prog. Aerosp. Sci. **45**(1–3), 50–79 (2009)
10. Snelson, E., Ghahramani, Z.: Sparse Gaussian processes using pseudo-inputs. In: Advances in Neural Information Processing Systems, pp. 1257–1264 (2006)
11. Couckuyt, I., et al.: Blind Kriging: implementation and performance analysis. Adv. Eng. Softw. **49**, 1–13 (2012)
12. Rasmussen, C.E., Nickisch, H.: Gaussian processes for machine learning (GPML) toolbox. J. Mach. Learn. Res. **11**, 3011–3015 (2010)

Modeling Thermofluid Systems: An Approach Customized for Optimization

Jonas B. Weber and Ulf Lorenz

1 Introduction

Technical components are usually well optimized with regard to a certain operating point. Nevertheless, combining these optimized components does not guarantee an optimal system. Therefore, paying attention to a holistic design process reveals new potential. To make use of this potential, we investigate an algorithmic system design approach.

An extensively investigated topic in this regard and the foundation of this work are flow networks [1]—more specifically, fluid systems [2]. However, in addition to the transport of fluid, many technical applications involve heating and cooling. The corresponding systems are then called thermofluid systems.

In this context, Pöttgen et al. [3] examined and optimized an already existing heating circuit of a conference building. In this paper, however, we present a mixed-integer linear program for the system synthesis task of thermofluid systems. While the first presented quasi-stationary representation can be reasonable for some systems, it does not hold for the general case. Due to the use of storage tanks, dynamic behavior occurs which demands for an appropriate time representation. We therefore introduce a continuous-time based extension to model these dynamic effects.

J. B. Weber (✉) · U. Lorenz
Chair of Technology Management, University of Siegen, Siegen, Germany
e-mail: jonas.weber@uni-siegen.de; ulf.lorenz@uni-siegen.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_48

2 Thermofluid Systems

The system synthesis task considered in this paper can be stated as follows: Given a construction kit of technical components as well as a technical specification of load collectives, compare all systems which are able to satisfy every prospected load and choose the one with the minimal lifespan costs.

A possible representation for this purpose is to model a system as a source-target-network (G, s, t) with directed multigraph $G := (V, E)$. In this context, the set of edges E represents technical components and the set of vertices V represents interconnections between components, whereas $s, t \in V$ are two distinguished vertices, namely the source and the target of the network. Each possible system, defined by a purchase decision of components, can be represented as a connected subgraph without directed cycles.

In general, thermofluid systems can be described as fluid systems with superimposed heat transfer. However, the pumping constraints are not presented in the following due to space limitations. For a more detailed version of the model, we refer to [4]. Note that for the quasi-stationary case considered here, similar loads of a load collective are aggregated to so-called load scenarios S, each of them occurring for a specific portion of a system's lifespan. An overview of all variables and parameters used is given in Table 1. All non-linear constraints are linearized using the techniques presented in [5].

$$\min \quad c_{invest} + c_{energy} \tag{1}$$

$$a^s_{i,j} \leq b_{i,j} \qquad \forall s \in S,\ (i,j) \in E \tag{2}$$

$$\dot{v}^s_{i,j} \leq \dot{V}^{max} \cdot a^s_{i,j} \qquad \forall s \in S,\ (i,j) \in E \tag{3}$$

$$\dot{q}^{in\,s}_{i,j} \leq \dot{Q}^{max} \cdot a^s_{i,j} \qquad \forall s \in S,\ (i,j) \in E \tag{4}$$

$$\dot{q}^{out\,s}_{i,j} \leq \dot{Q}^{max} \cdot a^s_{i,j} \qquad \forall s \in S,\ (i,j) \in E \tag{5}$$

$$\sum_{(i,k)\in E} \dot{v}^s_{i,k} - \sum_{(k,j)\in E} \dot{v}^s_{k,j} = 0 \qquad \forall s \in S,\ k \in V \setminus \{s,t\} \tag{6}$$

$$\sum_{(i,k)\in E} \dot{q}^{out\,s}_{i,k} - \sum_{(k,j)\in E} \dot{q}^{in\,s}_{k,j} = 0 \qquad \forall s \in S,\ k \in V \setminus \{s,t\} \tag{7}$$

$$t^s_k = T\Big(\sum_{(i,k)\in E} \dot{v}^s_{i,k}, \sum_{(i,k)\in E} \dot{q}^{out\,s}_{i,k}\Big) \qquad \forall s \in S,\ k \in V \setminus \{s,t\} \tag{8}$$

$$T(\dot{v}^s_{i,j}, \dot{q}^{in\,s}_{i,j}) \leq t^s_i + (1 - a^s_{i,j}) \cdot T^{max} \qquad \forall s \in S,\ (i,j) \in E \tag{9}$$

$$T(\dot{v}^s_{i,j}, \dot{q}^{in\,s}_{i,j}) \geq t^s_i - (1 - a^s_{i,j}) \cdot T^{max} \qquad \forall s \in S,\ (i,j) \in E \tag{10}$$

$$\dot{q}^{out\,s}_{i,j} \leq \dot{q}^{in\,s}_{i,j} + \Delta\dot{q}^s_{i,j} + (1 - a^s_{i,j}) \cdot \dot{Q}^{max} \qquad \forall s \in S,\ (i,j) \in E \setminus TS \tag{11}$$

Table 1 Variables and parameters for the quasi-stationary model

Symbol	Range	Description
$b_{i,j}$	$\{0; 1\}$	Purchase decision of component (i, j)
$a^s_{i,j}$	$\{0; 1\}$	Activation decision of component (i, j) in scenario s
$\dot{v}^s_{i,j}$	$\mathbb{R}^+_0$	Volume flow through component (i, j) in scenario s
$\dot{q}^{in\ s}_{i,j}$	$\mathbb{R}^+_0$	Heat flux at the inlet of component (i, j) in scenario s
$\dot{q}^{out\ s}_{i,j}$	$\mathbb{R}^+_0$	Heat flux at the outlet of component (i, j) in scenario s
t^s_k	$\mathbb{R}^+_0$	Temperature at connection k in scenario s
$\Delta\dot{q}^s_{i,j}$	$\mathbb{R}$	Heat increase by heat source (i, j) in scenario s
$t^s_{i,j}$	$\mathbb{R}^+_0$	Outlet temperature of temp. source (i, j) in scenario s
$\dot{V}^{max}$	–	Upper bound on the volume flow
$\dot{Q}^{max}$	–	Upper bound on the heat flux
T^{max}	–	Upper bound on the temperature

$$\dot{q}^{out\ s}_{i,j} \geq \dot{q}^{in\ s}_{i,j} + \Delta\dot{q}^s_{i,j} - (1 - a^s_{i,j}) \cdot \dot{Q}^{max} \qquad \forall s \in S,\ (i, j) \in E \backslash TS \tag{12}$$

$$t^s_j \leq t^s_{i,j} + (1 - a^s_{i,j}) \cdot T^{max} \qquad \forall s \in S,\ (i, j) \in TS \tag{13}$$

$$t^s_j \geq t^s_{i,j} - (1 - a^s_{i,j}) \cdot T^{max} \qquad \forall s \in S,\ (i, j) \in TS \tag{14}$$

The objective of the presented model is to minimize the sum of investment costs and expected energy costs over a system's lifespan, see (1). A component can only be used to satisfy a load scenario if it is installed, see (2). If a component is operational, its volume flow and heat flux are reasonable or vanish otherwise, see (3)–(5). Due to the law of flow conservation, the volume flow and heat flux have to be preserved at all vertices, except for the source and the target, see (6) and (7). For the mixture of incoming flows at the vertices, except for the source and the target, the resulting temperature depends on the sums of the incoming volume flows and heat fluxes, see (8). Furthermore, all flows exiting an operational component must have the same temperature, see (9) and (10). If a heating (or cooling) component is operational, the transferred heat increases (or decreases) the heat flux, see (11) and (12). For non-heating (or -cooling) components this increase (or decrease) is typically 0. However, there is an exception as we differentiate between two ideal sources of thermal energy: ideal heat sources and ideal temperature sources. While an ideal heat source delivers a constant heat flux, ideal temperature sources maintain a constant outlet temperature. In the case of ideal temperature sources denoted as set TS, Constraints (11) and (12) do not apply. Rather a constant temperature is assigned to an operational component's outlet, see (13) and (14).

3 Continuous-Time Representation

The presented quasi-stationary modeling approach is not applicable for systems with storage, existing components with extensive start-up and run-down phases, general delayed system responses or the like. For these systems, similar loads cannot be aggregated to scenarios since the state of a system at a certain point in time depends on the load history. Therefore, an appropriate time representation which meets this requirement must be developed. It should be noted that the approach presented in this section focuses on the application of storage tanks, although it may be adoptable for other purposes, too.

For the representation of time, two contrary types exist—discrete and continuous ones. Due to the disadvantages of discrete representations [4], we focus on a continuous-time representation with global events. This means that the system events (or actions) define a joint, unified reference grid for all components of the system. The basic idea, however, is that (additional) variables are used to determine the timings of time intervals. As a consequence, the main difficulty of this approach is to estimate and adjust the number of time intervals required.

The demand of a given system changes when an activity induced by a consumer, i.e. a sink, takes place or the demand varies indirectly due to the filling or emptying of storage tanks. The change in demand due to the first is called a main-event and is specified in advance. Hence, the challenge is to determine the number of intervals between these main-events.

For a constant demand at the sink, a storage tank should strive to empty itself as early and fill itself as late as possible during this period to avoid energy losses. Even if energy losses are not explicitly considered, it is reasonable to assume without loss of generality that the filling or emptying takes place in only one continuous process instead of multiple, interrupted ones right before or after a main-event. We can therefore define that at most one filling and one emptying process per tank takes place between two main-events. Thus, the upper bound on the number of intervals between two main-events is:

$$n_{intervals} = n_{sources} + 2 \cdot n_{tanks}$$

Consequently, in the case of one source and one storage tank, there are at most three intervals between two main-events. However, the determined number of intervals is only an upper bound. If for example the tank is neither filled nor emptied and the demand is solely satisfied by a continuous flow from the source to the sink, all intervals but the second exist only theoretically and have a length of 0—whereby, according to the presented considerations, the first interval is associated with emptying and the third interval with filling the tank.

In the following, the basic model for this approach is presented. The additional variables and parameters which are now required are shown in Table 2. Besides that, the model described above can be easily adapted for quasi-stationary thermofluid systems by replacing the load scenarios S with the corresponding main-events $\mathcal{E}$ and

Table 2 Additional variables and parameters for the dynamic model

Symbol	Range	Description
$\Delta\dot{v}_\tau^{e,l}/\Delta\dot{q}_\tau^{e,l}$	$\mathbb{R}$	Relative volume/heat change of storage tank (i, τ, j) in interval l for event e
$\Delta v_\tau^{e,l}/\Delta q_\tau^{e,l}$	$\mathbb{R}$	Absolute volume/heat change of storage tank (i, τ, j) in interval l for event e
$d^{e,l}$	$\mathbb{R}_0^+$	Duration of interval l for event e
$v_\tau^{begin\,e,l}/q_\tau^{begin\,e,l}$	$\mathbb{R}_0^+$	Volume/heat level of storage tank (i, τ, j) at the beginning of interval l for event e
$v_\tau^{end\,e,l}/q_\tau^{end\,e,l}$	$\mathbb{R}_0^+$	Volume/heat level of storage tank (i, τ, j) at the end of interval l for event e
$V_\tau^{init.}/Q_\tau^{init.}$	–	Initial volume/heat level of storage tank (i, τ, j)
V_τ^{max}	–	Maximum filling level of storage tank (i, τ, j)

intervals I. It should be noted that storage tanks are a special type of components as they can also act as (volume-restricted) sources or sinks. Tanks are, therefore, modeled as ordinary components (i, j) with an additional vertex $\tau \in A$ in between representing the property to store fluid (i, τ, j). Here, A is a new set of vertices with $A \cap V = \emptyset$.

$$b_{i,\tau} = b_{\tau,j} \qquad \forall\,(i,\tau),\ (\tau,j) \in E :\ \tau \in A \tag{15}$$

$$a_{i,\tau}^{e,l} + a_{\tau,j}^{e,l} \le 1 \qquad \forall\, e \in \mathcal{E},\ l \in I,\ (i,\tau),\ (\tau,j) \in E :\ \tau \in A \tag{16}$$

$$\sum_{(i,\tau)\in E} \dot{v}_{i,l}^{e,l} - \sum_{(\tau,j)\in E} \dot{v}_{\tau,j}^{e,l} = \Delta\dot{v}_\tau^{e,l} \qquad \forall\, e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{17}$$

$$\sum_{(i,\tau)\in E} \dot{q}_{i,\tau}^{out\,e,l} - \sum_{(\tau,j)\in E} \dot{q}_{\tau,j}^{in\,e,l} = \Delta\dot{q}_\tau^{e,l} \qquad \forall\, e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{18}$$

$$\Delta v_\tau^{e,l} = V(\Delta\dot{v}_\tau^{e,l},\, d^{e,l}) \qquad \forall\, e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{19}$$

$$\Delta q_\tau^{e,l} = Q(\Delta\dot{q}_\tau^{e,l},\, d^{e,l}) \qquad \forall\, e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{20}$$

$$v_\tau^{begin\,e,l} = V_\tau^{init.} + \sum_{\varepsilon\in\mathcal{E}:\varepsilon<e} \sum_{l\in I} \Delta v_\tau^{\varepsilon,l} + \sum_{\lambda\in I:\lambda<l} \Delta v_\tau^{e,\lambda} \qquad \forall\, e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{21}$$

$$v_\tau^{end\,e,l} = v_\tau^{begin\,e,l} + \Delta v_\tau^{e,l} \qquad \forall\, e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{22}$$

$$q_\tau^{begin\,e,l} = Q_\tau^{init.} + \sum_{\varepsilon\in\mathcal{E}:\varepsilon<e} \sum_{l\in I} \Delta q_\tau^{\varepsilon,l}$$

$$+ \sum_{\lambda \in I : \lambda < l} \Delta q_\tau^{e,l} \qquad \forall e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{23}$$

$$q_\tau^{end\ e,l} = q_\tau^{begin\ e,l} + \Delta q_\tau^{e,l} \qquad \forall e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{24}$$

$$v_\tau^{end\ e,l} \leq V_\tau^{max} \qquad \forall e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{25}$$

$$t_\tau^{e,l} = T(v_\tau^{begin\ e,l},\ q_\tau^{begin\ e,l}) \qquad \forall e \in \mathcal{E},\ l \in I,\ \tau \in A \tag{26}$$

For a tank either both of its edges or neither of them can be installed, see (15). Furthermore, a tank can either be filled or emptied but not both at a time, see (16). For a tank's storage vertex, the law of flow conservation has to be modified to model the relative volume and heat changes of the tank, see (17) and (18). The resulting absolute volume and heat change within an interval is then determined by the relative change as well as the duration of the interval, see (19) and (20). These changes can be further used to calculate the volume and heat levels of the tank at the beginning and end of each interval, see (21)–(24). Thereby, it has to be ensured that the tank's storage capacity is not exceeded, see (25). In contrast to regular vertices, the temperature at a storage vertex is defined by its content, see (26). In addition to these constraints, several other constraints apply but are not presented here due to space limitations. However, these constraints are straightforward to implement. They handle the right sequencing and timing of intervals, assign each action to exactly one interval (and vice versa) as well as ensure an ordering of filling and emptying processes between two main-events in accordance to the considerations made.

4 Conclusion and Outlook

In this paper, we presented an optimization model for the algorithmic system design of thermofluid systems. The model provides a unified framework for later work on technical applications in this field. A possible real-world application are temperature control systems which are used, inter alia, for injection molding.

Additionally, we introduced a continuous time-representation for technical fluid-based systems. The proposed representation enables the consideration of dynamic effects. These, in turn, are considered in such a way that the essential properties of storage tanks can still be taken into account while the representation is simple enough to be applicable for optimization. The approach, however, is particularly advantageous if there is a manageable number of load changes. Thus, a promising application could be to apply it to industrial processes with a regularly, e.g. daily, repeating production sequence.

References

1. Martin, A., Klamroth, K., Lang, J., Leugering, G., Morsi, A., Oberlack, M., Ostrowski, M., Rosen, R.: Mathematical Optimization of Water Networks. International Series of Numerical Mathematics, vol. 162. Birkhäuser, Basel (2012)
2. Pelz, P.F., Lorenz, U., Ederer, T., Lang, S., Ludwig, G.: Designing pump systems by discrete mathematical topology optimization: the artificial fluid systems designer (AFSD). In: International Rotating Equipment Conference, Düsseldorf (2012)
3. Pöttgen, P.F., Ederer, T., Altherr, L.C., Lorenz, U., Pelz, P.F.: Examination and optimization of a heating circuit for energy-efficient buildings. Energy Technol. **4**(1), 136–144 (2016)
4. Weber, J.B., Lorenz, U.: Algorithmic system design of thermofluid systems. arXiv:1806.09939 [math.OC] (2018)
5. Vielma, J.P., Ahmed, S., Nemhauser, G.: Mixed-integer models for nonseperable piecewise linear optimization: unifying framework and extensions. Oper. Res. **58**(2), 303–315 (2010)

Part XII
Pricing and Revenue Management

Data-Driven Stochastic Dynamic Pricing and Ordering

Rainer Schlosser

1 Introduction

Price updates on today's online markets increase in number and happen in shorter intervals. Merchants frequently adjust their prices to competitors' prices in order to increase profits. Electronic markets are ideal environments for dynamic pricing strategies because price changes are cheap and automatable. Software agents have fast response times and can analyze huge amounts of historical market data to estimate the buying behavior of customers.

Another important task of a merchant is to organize the inventory replenishment. If the inventory level is too low, the merchant might miss potential sales due to a stock-out. But storing too many items causes high holding costs. The ordering problem is about *when* to order *how many* items.

Managing prices and inventory is difficult because pricing and ordering decisions influence each other. The optimal price depends not only on (time-dependent) demand but also on the current inventory level. If the merchant reduces prices, the demand is likely to increase, which must be considered in the order decision. Because of this mutual influence, ordering and pricing should be decided jointly. Moreover, it is crucial to accurately predict future demand for good decisions.

The main contribution of this paper is the following:

- We use a data-driven approach to predict sales probabilities.
- We derive joint dynamic pricing and ordering strategies.
- We include inventory holding costs, ordering costs, and discounting.

R. Schlosser (✉)
Hasso Plattner Institute, Potsdam, Germany
e-mail: rainer.schlosser@hpi.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_49

This paper is organized as follows. In Sect. 2, we discuss related work. In Sect. 3, we describe the stochastic dynamic pricing and ordering model. In Sect. 4, we show how historical market data can be analyzed to estimate demand. In Sect. 5, we derive feedback pricing and ordering policies based on estimated sales probabilities. Potential extensions of our proposed model (particularly, competition and multi-product settings) are discussed in the final Sect. 6.

2 Related Work

Inventory control problems as well as dynamic pricing problems have been extensively studied, cf., e.g., Arrow et al. [3], Scarf [13] for pure ordering problems or [12, 18] for pure pricing problems.

Joint dynamic pricing and ordering problems are reviewed in the survey by Elmaghraby and Keskinocak [8]. Solutions are proposed for different problem scenarios, if demand is known, cf. [9], [19], or [17].

Scenarios with uncertain demand are less well studied. Expected future demand must be estimated from market observations. Typical approaches are to investigate specific classes of parameterized demand distributions in order to find parameters, so that the demand distribution fits the experienced sales best, cf. [4]. Bisi and Dada [5] propose Bayesian based approaches for ordering and pricings problem with uncertain demand. Adida and Perakis [1] study this problem in a multi-product scenario without backordering.

Further, in recent literature there are approaches to also incorporate competition. Chen and Chen [6] and Den Boer [7] provide an overview about dynamic pricing problems under competition for single-product and multi-product scenarios. Finite time horizon settings have been studied, e.g., by Martinez-de-Albeniz and Talluri [11]. Data-driven repricing strategies for infinite horizon oligopolies are derived in [15]. Adida and Perakis [2] consider joint pricing and inventory control in a duopoly.

The combined problem of (i) joint ordering and pricing, (ii) demand learning, and (iii) oligopoly competition is highly challenging. As optimal solutions are no longer tractable, heuristic approaches have to be used. For analyzing and evaluating the complex interplay of data-driven or rule-based ordering and pricing strategies under competition simulation platforms, cf. [16], can be used.

3 Model Description

We consider the situation in which a firm seeks to sell a durable good over time. We assume that (i) demand is uncertain and has to be estimated from historical data, (ii) prices can be adjusted over time, and (iii) items can be reproduced

or reordered. The goal is to derive data-driven pricing and ordering decisions to maximize expected discounted long-term profits.

If a sale takes place shipping costs c have to be paid, $c \geq 0$. A sale of one item at price a leads to profit of $a - c$. Moreover, we consider inventory holding costs. We assume that each unsold item leads to holding costs of l per period (e.g., 1 h or 1 day), $l \geq 0$. The time horizon is assumed to be infinite. For the length of one period, we use the discount factor δ, $0 < \delta < 1$.

We assume that demand depends on price. The set of admissible prices is denoted by A. Further, we allow sales probabilities to depend on time, e.g., the time of the day or the weekday. We assume that the time-dependence is periodic and has a finite cycle length of L periods. The probability to sell exactly i items within the time frame $(t, t+1)$ at price a is denoted by, $a \in A$, $i = 0, 1, 2, \ldots$, $t = 0, 1, 2, \ldots$,

$$P_t(i, a) \quad where \quad \forall t \quad P_{t \bmod L}(i, a) = P_k(i, a), \quad k = 0, 1, \ldots, L-1 \tag{1}$$

The random inventory level at time t is denoted by N_t, $t = 0, 1, \ldots$. If all items are sold, we let $P_t(\cdot, \cdot) = 0$ (no back orders). New items can be ordered at time $t = 0, 1, \ldots$. The number of items ordered at time t is denoted by b_t. The set of admissible orders quantities are denoted by B. Ordered products are assumed to be delivered with a delay of one period (e.g., 1 day), i.e., in time $t + 1$ the inventory level N_{t+1} raises by b_t. Ordering costs are paid in advance and given by, e.g., $b \in B$,

$$C(b) := c_{fix} \cdot 1_{\{b>0\}} + c_{var} \cdot b$$

using fixed and variable cost parameters $c_{fix}, c_{var} \geq 0$. For each period t, i.e., the interval $(t, t+1)$, $t = 0, 1, 2, \ldots$, an offer price a_t and an ordering quantity b_t has to be chosen. We call strategies $(a_t, b_t)_t$ admissible if they belong to the class of Markovian feedback policies; i.e., pricing $a_t > 0$ and ordering decisions $b_t \geq 0$ may depend on time t and the current inventory level N_t.

By X_t, we denote the random number of sales in period $(t, t+1)$, $t = 0, 1, 2, \ldots$. Depending on the chosen pricing/ordering strategy $(a_t, b_t)_t$, the random accumulated profit from time t on (discounted on time t) amounts to, $t = 0, 1, 2, \ldots$,

$$G_t := \sum_{s=t}^{\infty} \delta^{s-t} \cdot ((a_s(N_s) - c) \cdot X_s - l \cdot N_s - C\,(b_s(N_s))) \tag{2}$$

The objective is to determine a non-anticipating (Markovian) feedback pricing and ordering policy that maximizes the expected discounted total profit $E\,(G_0\,|N_0\,)$, where N_0 denotes the initial inventory level in $t = 0$. In the next Sect. 4, we show how sales probabilities can be estimated from historical market data. In Sect. 5, we show how to solve the optimization problem, cf. (1) and (2).

4 Estimation of Sales Probabilities

The goal of this section is to estimate sales probabilities from historical market data. The idea is to find a relation between the number of realized sales and offered prices within different time intervals. In the following, we assume that there is data for J time intervals, $j = 1, \ldots, J$. Data is supposed to consist of offer prices $a_{t^{(j)}}$ and realized sales $y_{t^{(j)}}$, i.e., the number of products sold in period $t^{(j)}$, $j = 1, \ldots, J$.

In the following, we show how to estimate sales intensities to calibrate our dynamic model, cf. Sect. 3. To explain the dependent variable $y_{t^{(j)}}$, $j = 1, \ldots, J$, we can use, e.g., a robust least squares regression model (LS model). Following the LS model, we aim to specify sales intensities $\lambda_t(a)$, $t = 0, 1, \ldots, L-1$, $a \in A$,

$$\lambda_t(a; \boldsymbol{\beta}) := \mathbf{x}(t, a)'\boldsymbol{\beta} \tag{3}$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)$ is the unknown parameter vector that is associated to the vector $\mathbf{x} = (x_1, \ldots, x_M)$ of M explanatory variables. The regressors $\mathbf{x}(t, a)$ can be a function of time t and the offer price a. The optimal coefficients $\boldsymbol{\beta}^* = (\beta_1^*, \ldots, \beta_M^*)$, can be easily obtained using standard methods.

Finally, the resulting intensities $\lambda_t^*(a) := \lambda_t(a; \boldsymbol{\beta}^*)$, cf. (3), are used to estimate sales probabilities: For period t, we let $P_t(\cdot, a)$ be Poisson distributed with rate $\lambda_t^*(a)$, $a \in A$, $t = 0, 1, \ldots, L-1$. Note, the time dependence of λ can be captured by time-dependent explanatory variables. To illustrate the approach, in the following definition, we give simple examples of explanatory variables $\mathbf{x}$.

Definition 1 We define the following regressors, $a_{t^{(j)}} \in A$, $t^{(j)} \geq 0$, $j = 1, \ldots, J$:

$x_1(t^{(j)}, a_{t^{(j)}}) = 1$	constant/intercept
$x_2(t^{(j)}, a_{t^{(j)}}) := a_{t^{(j)}}$	own price at time $t^{(j)}$
$x_3(t^{(j)}, a_{t^{(j)}}) := r_{t^{(j)}} - a_{t^{(j)}}$	price gap between $a_{t^{(j)}}$ and a reference price $r_{t^{(j)}}$
$x_{4+k}(t^{(j)}, a_{t^{(j)}}) := 1_{\{t^{(j)} \bmod L = k\}}$	time-effect/seasonality k, $k = 0, \ldots, L-1$

Our basic framework allows to measure the price and time dependence of average sales. Note, also non-linear versions of explanatory variables, cf. x_2, or certain reference prices (r_t), cf. x_3, can be used.

The goal of the next section is to find best pricing and ordering decisions by taking estimated demand as well as ordering and holding costs into account.

5 Dynamic Model and Solution Approach

In this section, we derive optimized pricing and ordering strategies. Recall, the time dependence in our infinite horizon model is assumed to be seasonal (daily/weekly effects) with a given cycle length of L periods. For all t, where $t \bmod L = k$, we

have $P_t(i,a) = P_k(i,a), k = 0, 1, \ldots, L-1, i = 0, 1, \ldots, a \in A$, cf. (1). The sales probabilities P can be calibrated using data-driven estimations, cf. Sect. 4.

Using dynamic programming, the best expected future profits $E(G_t|N_t = n)$, cf. (2), are described by the value function $V_t(n)$, $t, n = 0, 1, \ldots$. We can assume $V_t(n) = V_{t \bmod L}(n)$ for all t. Hence, we just have to determine the values $V_t(n)$, $t = 0, 1, \ldots, L-1$. If a period's demand is i and b items are ordered (with delivery delay), the transition of the current inventory level n is given by $n \to \max(n - i, 0) + b$. To avoid an unbounded state space, we use the upper limit N_{max}, which—if chosen sufficiently large—does not affect the optimal solution $(a_t^*(n), b_t^*(n))$, which is characterized by the argmax of the Bellman equation, $t = 0, 1, \ldots, L-1$, $n = 0, 1, \ldots, N_{max}$,

$$V_t(n) = \max_{a \in A, b \in B} \left\{ \sum_{i=0}^{N_{\max}} P_t(i,a) \cdot \begin{pmatrix} (a-c) \cdot \min(i,n) - l \cdot n - C(b) + \\ \delta \cdot V_{(t+1) \bmod L}\left(\min\left(\max(n-i,0) + b, N_{\max}\right)\right) \end{pmatrix} \right\} \tag{4}$$

If optimal prices or ordering quantities are not uniquely determined, we choose the largest numbers. The system (4) can be solved by solvers or using recursive dynamic programming techniques starting with a terminal condition, e.g., $V_T(\cdot) = 0$ for a sufficiently large T, where the approximation error can be estimated via the discount $\delta < 1$. Overall, our proposed framework is of low complexity and allows for adaptive demand estimations and frequent recomputations of strategies.

6 Extensions and Future Work

We have proposed a basic data-driven stochastic dynamic pricing and ordering model. Using estimated sales probabilities, we have set up a time-dependent framework including discounting, holding costs, and ordering costs. Our approach can be extended to problems with (i) multiple offer dimensions, (ii) competitive settings, and (iii) multiple product types.

To address markets with multiple offer dimensions (e.g., quality, ratings, shipping time, etc.) the demand learning component needs to be extended. In our setting, additional characteristic explanatory variables can be easily defined.

To include competition is challenging as competitors' offers dramatically increase the size of the state space. Further, competitors' strategies and their inventory levels are typically not observable. The estimation of sales probabilities has to include competitors' offers. A viable approach is to extend the data-driven single-product oligopoly model, cf. [15], by ordering decisions.

In multi-product models, the state space and the action space can be enormous. To manage this complexity, relaxation approaches and decomposition techniques can be used. The demand learning component has to be extended such that substitution effects are taken into account, cf. [10, 14].

In a combined model, private sales data, own offer data, and observable data of competitors' offers have to used to predict sales probabilities in competitive multi-product markets. In future research, we will combine the concepts described above to derive a model that allows to compute viable data-driven pricing and ordering strategies for real-life applications.

References

1. Adida, E., Perakis, G.: A robust optimization approach to dynamic pricing and inventory control with no backorders. Math. Program. **107**(1–2), 97–129 (2006)
2. Adida, E., Perakis, G.: Dynamic pricing and inventory control: uncertainty and competition. Oper. Res. **58**(2), 289–302 (2010)
3. Arrow, K.J., Harris, T., Marschak, J.: Optimal inventory policy. Econometrica **19**(3), 250–272 (1951)
4. Azoury, K.S.: Bayes solution to dynamic inventory models under unknown demand distribution. Manag. Sci. **31**(9), 1150–1160 (1985)
5. Bisi, A., Dada, M.: Dynamic learning, pricing, and ordering by a censored newsvendor. Naval Res. Logist. **54**(4), 448–461 (2007)
6. Chen, M., Chen, Z.-L.: Recent developments in dynamic pricing research: multiple products, competition, and limited demand information. Prod. Oper. Manag. **24**(5), 704–731 (2015)
7. Den Boer, A.V.: Dynamic pricing and learning: historical origins, current research, and new directions. Surv. Oper. Res. Manag. Sci. **20**(1), 1–18 (2015)
8. Elmaghraby, W., Keskinocak, P.: Dynamic pricing in the presence of inventory considerations: research overview, current practices, and future directions. Manag. Sci. **49**(10), 1287–1309 (2003)
9. Federgruen, A., Heching, A.: Combined pricing and inventory control under uncertainty. Oper. Res. **47**(3), 454–475 (1999)
10. Ito, S., Fujimaki, R.: Optimization beyond prediction: prescriptive price optimization. In: 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2017), pp. 1833–1841 (2017)
11. Martínez-de-Albéniz, V., Talluri, K.T.: Dynamic price competition with fixed capacities. Manag. Sci. **57**(6), 1078–1093 (2011)
12. Phillips, R.L.: Pricing and Revenue Optimization. Stanford University Press, Palo Alto (2005)
13. Scarf, H.E.A.: Survey of analytic techniques in inventory theory. Multistage Inventory Models Tech. **7**, 185–225 (1963)
14. Schlosser, R.: Stochastic dynamic multi-product pricing under competition. Oper. Res. Proc. **2017**, 527–533 (2018)
15. Schlosser, R., Boissier, M.: Dynamic pricing under competition on online marketplaces: a data-driven approach. In: 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2018), pp. 705–714 (2018)
16. Schlosser, R., Walther, C., Boissier, M., Uflacker, M.: Data-driven inventory management and dynamic pricing competition on online marketplaces. In: 27th International Joint Conference on Artificial Intelligence (IJCAI 2018), pp. 5856–5858 (2018)
17. Simchi-Levi, D., Chen, X., Bramel, J.: Integration of inventory and pricing. In: The Logic of Logistics. Springer, Berlin (2014)

18. Talluri, K.T., van Ryzin, G.: The Theory and Practice of Revenue Management. Kluwer Academic Publishers, Boston (2004)
19. Transchel, S., Minner, S.: The impact of dynamic pricing on the economic order decision. Eur. J. Oper. Res. **198**(3), 773–789 (2009)

Part XIII
Production and Operations Management

Tactical Planning of Modular Production Networks in the Chemical Industry: A Case Study in Specialty Polymers

Tristan Becker, Bastian Bruns, Stefan Lier, and Brigitte Werners

1 Introduction

Faced with new challenges, the chemical industry strives for more diversified product portfolios. Product life cycles are becoming shorter and lead to an increased uncertainty in product demands with regard to volume, location and type. To efficiently cope with highly individual demands, modular production concepts are considered. Modular plants can be installed in standard transportation containers and may operate at decentralized facilities in direct proximity to customers or suppliers. New modular plants can be added and existing plants can be relocated in the short-term to adapt to changing demand conditions. For an efficient utilization of the increased flexibility provided by modular plants, tactical planning must incorporate the new flexibility options. In this paper, using real data provided by a large chemical company, the economic benefits of modular production concepts in the specialty polymers market are assessed. The paper is organized as follows: in Sect. 2 the opportunities of modular production concepts for the case of specialty polymers are pointed out. A novel model formulation for the tactical network

T. Becker (✉) · B. Werners
Ruhr University Bochum, Faculty of Management and Economics, Chair of Operations Research and Accounting, Bochum, Germany
e-mail: tristan.becker@rub.de

B. Bruns
Ruhr University Bochum, Institute for Thermo- and Fluiddynamics, Department of Mechanical Engineering, Laboratory of Fluid Separations, Bochum, Germany
e-mail: bruns@fluidvt.rub.de

S. Lier
University of Applied Sciences Südwestfalen, Department of Engineering and Economics, Iserlohn, Germany
e-mail: Lier.Stefan@fh-swf.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_50

planning with relocatable plants for the specialty chemical industry is presented in Sect. 3. In Sect. 4 the favorability of modular production concepts is discussed comparing a realistic case with global and local sourcing options, which are enabled by decentral production. Section 5 concludes with a brief conclusion and directions for future research.

2 Opportunities of Modular Plants for the Production of Specialty Polymers

Small scale modular production units can provide various advantages compared to large-scale production plants [6]. Due to the design of modular production units, which can be installed in an ISO transportation container, it is easy to shift production facilities in the short-term enabling increased location flexibility. As modular plants may be operated at facilities that are established in the short term, modular plants can be placed in close proximity to customers or resources. This enables new opportunities regarding the supply chain and network structure. Thus, the production can be adapted to local demand. If the demand shifts regionally, modular production plants can easily be transported to another region or country. Decentralized production can potentially reduce transportation distances, leading to reduced logistics costs and CO_2 emissions. Additionally, modular production plants provide flexibility regarding the production volume. By numbering up or down of containers it is possible to produce different amounts of product if local demand increases or decreases. The production capabilities of modular plants can be adjusted by exchanging apparatuses, so that modular plants can adapt to changing demand type. Furthermore, the use of standardized, predetermined apparatuses in the size of laboratory equipment can greatly reduce the production time compared to conventional large scale plants. This leads to shorter time to market.

In this paper a process is considered which is the last step of the production of specialty polymers. This type of specialty polymers is produced by mixing a base polymer with a number of additives. As production inputs, in our case three base polymers with different viscosities are used. By mixing with different additives, the characteristics of the base polymers can be adjusted in terms of colour, softening and the ability of resisting ultraviolet light. The process consists of tank modules, a production module and a cleaning module which serve specific tasks. The required base polymer is pumped from tanks to the production module. In the production module the base polymer is heated up to a specific temperature. When the process temperature is reached, the base polymer is mixed with the required additives, which are pumped from barrels, to obtain the desired characteristics for the finished product. Pipes and mixers can be cleaned with a solvent, which can be regenerated in the cleaning module afterwards. Regeneration takes place in a membrane module within the cleaning module [4].

Furthermore, modular production concepts for the case of specialty polymers offer product flexibility. By installing multiple parallel process lines in each production module, different products or variations of products can be produced without changing the equipment. Thus, modular plants are able to produce different specialties by varying the mixture of base polymers and additives to serve different purposes [4]. This enables modular production plants to adapt to changing demands with regard to product type.

3 Tactical Planning of the Specialty Polymer Modular Production Network

Mathematical formulations for location problems with modular plant capacities have been investigated in the literature [3, 5]. The influence of demand uncertainties and the option of plant reconfigurations has been discussed by [1, 2]. In this section a model formulation for tactical network planning with relocatable plants in the specialty chemicals market is proposed. It serves as a basis for the discussion in our case study. The production and delivery of specialty polymers in the market under consideration is characterized by a make-to-order setting. In the following, a mixed-integer programming model is presented to minimize the network and production cost for fulfillment of all customer demand within the next year. Over the planning horizon, 12 months are considered as planning periods $t \in T$. There are three different product families $p \in P$, that are differentiated by the base polymer. Each modular plant is equipped with process lines for all product families that are operated independently and can produce $Cap_p^{prod} = 1000$t of each product per month. A number of 100 potential production facilities $i \in I$ has been selected. Each represents an industrial or chemical park where modular plants can be installed on a month by month basis. The set of 100 customers $j \in J$ is defined by the data that has been provided by a chemical company involved in the production of specialty polymers in the European region. Variable $x_{ijpt} \geq 0$ captures the amount of product p shipped from facility i to customer j in period t. The number m of modular plants at location i in period t is defined by $y_{imt} \in \{0, 1\}$. A facility i is operated in period t iff $\sum_{m \in M} y_{imt} = 1$. The number of plant relocations from facility i to i' in period t are denoted by variable $r_{ii't} \in \mathbb{N}$. Initial experiments have shown the following formulation to be well suited for network design of relocatable modular plants for specialty polymers:

$$\begin{aligned} \min \sum_{i \in I} \sum_{j \in J} \sum_{p \in P} \sum_{t \in T} c_{ijp}^{prod} x_{ijpt} + \sum_{i \in I} \sum_{m \in M} \sum_{t \in T} (c_i^{loc} + c_i^{op}) y_{imt} \\ + \sum_{i \in I} \sum_{i' \in I} \sum_{t \in T} c_{ii'}^{reloc} r_{ii't} \end{aligned} \tag{1}$$

$$\sum_{i \in I} x_{ijpt} = D_{jpt} \qquad \forall j \in J, p \in P, t \in T \tag{2}$$

$$\sum_{j \in J} x_{ijpt} \leq Cap_p^{prod} \sum_{m \in M} y_{imt} \qquad \forall i \in I, t \in T, p \in P \tag{3}$$

$$\sum_{m_1 \in M} m_1 y_{im_1 t} - \sum_{m_2 \in M} m_2 y_{im_2(t-1)} = \sum_{i \in I} (r_{i'i(t-1)} - r_{ii'(t-1)}) \quad \forall i \in I, t \in T \tag{4}$$

$$\sum_{m \in M} y_{imt} \leq 1 \qquad \forall i \in I \cup \{0\}, t \in T \cup \{0\} \tag{5}$$

$$y_{0|M|0} = 1, y_{im0} = 0, \sum_{i'} r_{ii'0} = 0 \qquad \forall i \in I, m \in M \tag{6}$$

The objective (1) minimizes the total network and production costs. Given the set of suppliers for raw material $r \in R$, let the street distance between each supplier and facility and each facility and customer be defined by d_{ri}^{raw} and d_{ij}^{cust} respectively. Additives are supplied from a single facility and the distance to each production facility is defined by d_i^{add}. The cost of transportation by truck per ton and kilometer of transportation is defined by c^{tp}. Each ton of finished product consists of 98% of base polymers and 2% of additives. Additives are specialized chemicals that cannot be sourced externally, while base polymers represent a commodity that can be sourced locally. The variable production cost per ton of product p are defined by c_p^{var}. Considering external sourcing of base polymers, the cost for raw material delivery to each location i, production and delivery of finished product p to each customer j can be defined as follows:

$$c_{ijp}^{prod} = 0.98 * c^{tp} \min\{d_{ri}^{raw},\ r \in R\} + 0.02 * c^{tp} * d_i^{add} + c_p^{var} + c^{tp} d_{ij}^{cust} \qquad \forall i \in I, j \in J, p \in P \tag{7}$$

Cost c_{ic}^{op}, c_i^{loc} and $c_{ii'c}^{reloc}$ denote the operational cost per modular plant at facility i, the operational cost of facility i for a single period and the relocation cost of modular plants respectively. Constraints (2) ensure all customer demand D_{jpt} is met. Constraints (3) ensure that the production does not violate the production capacity Cap_p^{prod} installed at each facility. As modular plants can be relocated between facilities, the number of modular plants installed at each facility is captured by constraints (4). If the number of modular plants deviates from one period to another, the difference has to be covered by relocations. Relocated modular plants

are available at the destination facility with a delay of one period. Constraint (5) ensures that only one capacity level may be chosen, which denotes the number of modular plants installed at each location in each period. Finally, facility $i = 0$ is the modular hub, where modular plants are assembled and initialized. No production takes place at location $i = 0$. In period $t = 0$, there are no demands and the initially available number of modular plants $|M|$ is deployed to the network from the modular hub (Constraints (6)).

4 Case Study

To investigate the favorability of modular production concepts for the case of specialty polymers in the chemical industry, a case study is conducted. The network and production costs of demand fulfillment over the course of a year are analyzed. Modular plants are available at the modular hub and can be deployed to the decentral facilities. The market for base polymers is typically characterized by production using large-scale plants and a strong competition. If specialty chemicals are produced at a central location, base polymers should also be sourced centrally. Logistic costs have a high share of the cost for specialty polymer production and delivery. As modular plants can be operated at decentral facilities, local sourcing of base polymers is considered as an option. Using the proposed model formulation and the standard solver Gurobi 8.0, both cases of global and local sourcing could be solved to optimality in a reasonable time. It turns out that there are no benefits associated with the utilization of modular plants in case of global sourcing. All production occurs at the central facility in close proximity to the source using four modular plants and no decentral facilities are opened. No cost for raw material transportation are incurred this way. The average distance of transportation per ton of product delivered is 1199.62 km (Fig. 1, left side). When external sourcing of

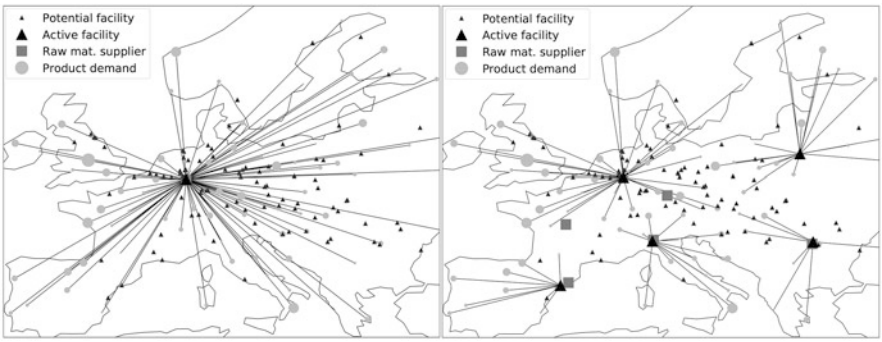


Fig. 1 Exemplary excerpt of network structure with global (left side) and local (right side) sourcing

Table 1 Comparison of cost in case of central and local sourcing

Type of costs (costs in million euros)	Global sourcing	Local sourcing
Raw material handling	0.00	0.36
Delivery of finished products	12.68	7.44
Variable production cost	5.28	5.28
Operational cost for modules	0.24	0.36
Operational cost for facilities	0.36	1.80
Cost for plant setup and relocations	0.06	0.22
Total	18.93	14.46

base polymers is taken into consideration, the structure of the production network is drastically altered. A number of five facilities and six modular plants are operated and the average distance per ton of product delivered is reduced to 703.8 km including the transportation of raw materials to the facilities (Fig. 1, right side). Production and network costs are reduced by 24% if local sourcing options are utilized (Table 1). In the network structure it can be observed that modular plants are positioned near to the local suppliers and customers and the cost of raw material handling is low. Figure 1 compares the network structure of the optimal solution in case of central and local sourcing. Delivery costs of finished product are reduced by 41.32% compared to global sourcing. Variable production cost are the same, since the same amount of customer demands is satisfied in both cases. The cost for the operation of facilities, modules and the cost for the setup and relocation of modules are increased by 360.61% in case of local sourcing compared to global sourcing. The increased cost are due to the higher number of modular plants and facilities that are operated. Operating additional facilities is costly, since a number of employees is required at each facility for maintenance and logistics. As shown in Fig. 1, for this application it is only beneficial to operate multiple facilities when local sourcing options are available.

5 Conclusion and Outlook

The implementation of modular production networks enables new opportunities with regard to customer orientation and flexibility of the production networks for the chemical industries. The case study confirms the economic benefits of a decentral modular production network with the possibility of local sourcing raw materials. Furthermore, the average distance of transportation of each ton of raw material and finished product is strongly reduced. This reduces the amount of CO_2 emissions associated with the logistics of the production network. An autonomous operation of modular plants could further reduce the operational cost for each facility so that a further reduction of logistics cost is possible. The share of the overall value chain for the production of specialty polymers is low. If local sourcing options are exploited

with decentral modular plants, the margin of base polymers may be lost and it is thus important to consider the margin of base polymers to evaluate the favorability for an individual company. Future research should focus on strategic planning of modular production networks and further applications with different flexibility options.

Acknowledgements This work was supported by the Deutsche Forschungsgemeinschaft (DFG) [grant number 387857236].

References

1. Becker, T., Lier, S., Werners, B.: Tactical planning of modular production networks with reconfigurable plants. In: Kliewer, N., Ehmke, J.F., Borndörfer, R. (eds.) Operations Research Proceedings 2017, pp. 549–555. Springer, Cham (2018)
2. Becker, T., Lutter, P., Lier, S., Werners, B.: Optimization of modular production networks considering demand uncertainties. In: Fink, A., Fügenschuh, A., Geiger, M.J. (eds.) Operations Research Proceedings 2016, pp. 413–418. Springer, Cham (2018)
3. Jena, S.D., Cordeau, J.f., Gendron, B.: Dynamic facility location with generalized modular capacities. Transp. Sci. **49**(3), 484–499 (2015)
4. Lier, S., Wörsdörfer, D., Paul, S., Ferdinand, D.: Schlusssachbericht ModuLOG: masterplan - logistik & supply chain strategien für modulare chemie- und pharmaproduktionen. Tech. Rep., Lehrstuhl für Fluidverfahrenstechnik, Ruhr-Universität Bochum (2015)
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)
6. Clausen, U., Langkau, S., Goedicke, I., Lier, S. Location and network planning for modular container plants in the process industry. Chem. Eng. Technol. (accepted author manuscript). https://doi.org/10.1002/ceat.201600735

Validating Measurement Data in Manufacturing Processes

David Brück and Sven Oliver Krumke

1 Introduction

In 1997, Hirsch-Kreinsen [1] hinted to an upcoming trend regarding manufacturing processes, which was designed to increase overall efficiency. Inspection processes in manufacturing traditionally consisted of a clear separation between the producing employee and the inspector. Said trend aimed to unite these responsibilities in one person, thus creating the so-called worker self-inspection method. Nowadays approximately 75% of manufacturers perform inspections in accordance with this method [2, 3]. However, back in 1997, Hirsch-Kreinsen already indicated that worker self-inspections would facilitate a greater likelihood of quality data manipulations.

The key requirement for Statistical Process Control (SPC) is possessing precise and trustworthy measurement data. If the underlying measurements are erroneous, all future analysis conclusions are superfluous. Thus, the whole process of monitoring and controlling becomes redundant and possible incorrect parameters and settings are overlooked. This disguises defective or faulty products, which in turn cause customer complaints that result in often considerable complaint handling costs and harm a company's reputation. Even though an experienced quality manager could find such manipulations by double checking the measurements, there is often simply no time to do so. In larger companies, thousands of measurements are gathered daily and managed via a Computer Aided Quality (CAQ) software system. Our contribution to this topic is to provide a tool that identifies samples that are more likely to have been manipulated. Using said tool, a quality manager can filter efficiently and only double check those few measurements that appear suspicious.

D. Brück (✉) · S. O. Krumke
Technische Universität Kaiserslautern, Kaiserslautern, Germany
e-mail: David.Brueck@CAQ.de; Krumke@mathematik.uni-kl.de;
http://www.mathematik.uni-kl.de/opt/mitglieder/krumke/

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_51

This work can be summarized in three points. Firstly, in Sect. 2, we characterize typical manipulations during manufacturing processes as patterns in a Measurement-Time-Series. Secondly, in Sect. 3, we summarize the algorithms and adjustments that were used to identify said patterns. Our final result is a combination of Decision Stump Forests and a novel variation of the Smith-Waterman algorithm which is able to operate with continuous data. Lastly, we highlight examples where real data from manufacturers in Germany was used to test the tool. All results displayed here are based on [4].

2 Characteristics of Typical Manipulations in Manufacturing Processes

It is well-known that typical manipulations can be performed very easily. In fact, typical manipulations can often be performed as easily as simply following the measurement instructions. In the rarest of cases, manipulations are intended to specifically harm a company. In general, these manipulations occur due to a lack of comprehension of why quality data is actually measured or in order to save time or reduce workload. If all measured products are "fine", one does not need to adjust parameters or rework erroneous products.

Together with some experts in quality management, we characterized typical manipulations on a Time-Series-Level via the following five different patterns:

1. Same value appears consecutively across several measurements,
2. Same distance to previous measurement appears consecutively across several measurements,
3. (Parts of) samples are copies of (Sub-)Sequences of previous samples,
4. (Parts of) samples are (Sub-)Sequences of previous samples, shifted by adding a constant $x \in \mathbb{R}$ to each measurement,
5. A sample "clings" to a certain point.

Pattern 5 will inevitably consist of a combination of the first four patterns. We therefore only focused on detecting the first four. After looking at these patterns and the underlying context of why this particular pattern is so unlikely to occur naturally, we can separate them into two distinct groups: spatially related and nonspatially related patterns. The first two patterns seem to be very implausible due to what appears to be a non-characteristic relationship between immediately consecutive values. The next two seem to be implausible because this exact pattern (or some pattern closely related to this) has already appeared at some point before.

3 Solution Architecture

Spatially related manipulation patterns are easier to detect since we already know what we're looking for. In order to deal with this problem, we trained a Decision Stump Forest via AdaBoost [5] to identify these patterns within small samples of size 3. Due to a greatly varying sample-size in practical usage, we use an overlapping sliding window approach to deal with this problem. This means that we do not have to train Decision Stump Forests for all possible sample-sizes.

Nonspatially related patterns are harder to detect, as we do not know exactly what we're looking for. Even though, in the third pattern, we are looking for exact copies of the tested sample, we, for instance, do not know how many measurements were copied. We also do not know where they were copied from or whether they are merely small copied parts which are followed by real measurement parts which are in turn continued by further copied parts. This gets even harder when we're looking for copies that were shifted in accordance with a fixed but unknown constant $x \in \mathbb{R}$.

The Measurement-Time-Series of a product is essentially the entire production processes' "DNA". Each measurement can be seen as a single nucleotide in a DNA-Sequence. We can therefore perform a type of comparative family relationship test between the new incoming sample and the existing historical data, represented by the ensemble of all previous samples. In case of a close relationship, we conclude that the new sample has a high chance of having been copied. To design that kind of test, we adjusted the Smith-Waterman algorithm [6] with affine gap costs [7] to also operate on continuous data.

In the original Smith-Waterman algorithm, one tries to find segments with maximum similarity between two molecular sequences $A = a_1 a_2 \dots a_n$ and $B = b_1 b_2 \dots b_m$, while a_i and b_j are taken from finite sets. A so called *similarity matrix* is used to value the similarity of two elements. It contains values for all possible pairs of elements a and b. It is obviously impossible to create a matrix like this for continuous a and b. We exchanged this similarity matrix with a *similarity function*, that rewards closely related a and b and punishes greater value distances. A proper approach to this is to normalize the data, e.g. via feature scaling $X' = \frac{X - X_{min}}{X_{max} - X_{min}}$ and use a typical reward function like this one $s(a, b) = (1 - |a - b|^3)^3$. However, a much easier approach surprisingly turned out to work best in our cases. The following *similarity function* $s(a, b)$ and affine gap costs $g(l)$, where l denotes the length of the corresponding gap, were used in our tests:

$$s(a, b) = \begin{cases} +6, & \text{if} a = b, \\ -8, & \text{else.} \end{cases} \tag{1}$$

$$g(l) = -5 - 2(l - 1) \tag{2}$$

In order to deal with the pattern of shifted subsequences, we needed our data or similarity function to be invariant to shiftings. This invariation is obtained by replacing a datapoint X_i with its distance to its previous point, while still maintaining the relative direction by setting $X'_i = X_i - X_{i-1}$.

Because measurements follow a random distribution, certain patterns will inevitably occur from time to time. Just because a single measurement exactly meets the nominal value, this alone does not necessarily indicate a manipulation. However, if this occurs suspiciously often throughout one sample, an increased likelihood of prior data manipulation exists. This realization is helpful, because in order to successfully "cheat", an inspector will not merely manipulate one single value, but a "convincing" number of values. We therefore compare the number of detected manipulations with the sample-size and set an applicable threshold ratio. If this threshold is not met, we delete all detected manipulations within the sample, thus reducing false positives.

4 Tests

Figure 1 shows the results we gathered by applying our validation algorithm to two different products from different manufacturers in Germany. Both results were shown to the respective heads of quality management and they both confirmed that the marked measurements had actually been manipulated. In both cases, the sample-size had been set to 6. We marked these samples that were flagged as "suspicious" with black rectangles or ellipses.

There are several things to note in Fig. 1. First of all, the range of possible values varies greatly from one Measurement-Time-Series to another. This is due to different products, different measurement characteristics, varying precision of measurement devices, etc. The top Time-Series contains a total of 360 single measurements. Only one sample has been flagged as "suspicious" due to a missing variance in comparison with all other samples. In this case, a quality manager would have had to only double check that one sample, instead of monitoring the whole

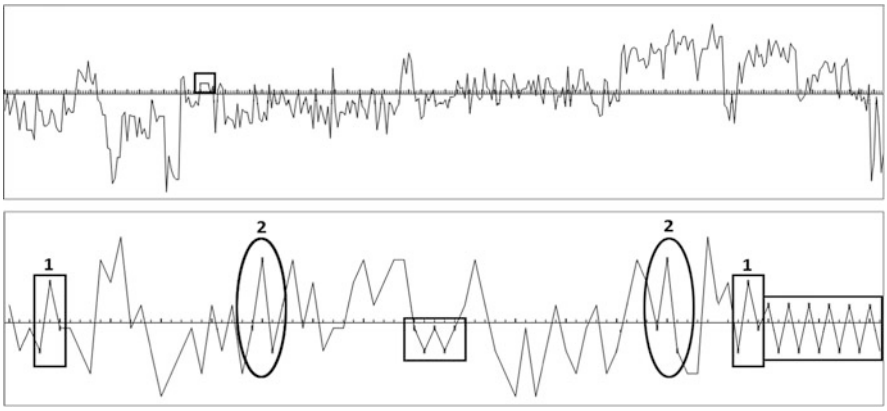


Fig. 1 Results of validation of two example time-series

Table 1 Mean ratio of different patterns occurring by chance

Hist. data len.	Manip.	≥ 3	≥ 4	≥ 5
~100	Repeats	0.44	0.03	0.002
	SameDist.	13	1.8	0.24
	CopySeq.	8.57	0.34	0.02
	CopyPat.	42.66	2.1	0.05
~350	Repeats	0.35	0.02	0.001
	SameDist.	11.8	1.38	0.16
	CopySeq.	0.73	0.01	0
	CopyPat.	19.77	0.32	0.01
~2000	Repeats	0.23	0.01	<0.001
	SameDist.	9.52	0.91	0.09
	CopySeq.	34.23	1.49	0.07
	CopyPat.	96.67	18.64	0.5

process. In contrast to this, the bottom example contains 84 measurements. Here, we mainly found two different types of suspicious patterns. The first one, highlighted with the numbers 1 and 2, respectively, show copied sequences. The unnumbered samples were marked as "suspicious" due to alternating values.

In order to approximate the likelihood of any "suspicious" pattern to occur merely by chance, we simulated new measurements. We took some already existing real Measurement-Time-Series from manufacturers in Germany and created new samples of size 6 by randomly picking and concatenating values of that existing Time-Series. Applying this method meant that it was impossible to create new samples with values that had never occurred before and the actual chance of specific patterns occurring randomly is overestimated. We looked for Time-Series of approximately the same length and then averaged our results. This allowed us to allocate estimates of differently sized groups of 100, 350, and 2000 values. After creating new samples of size 6, we checked for consecutive suspicious patterns of the same type. Table 1 depicts the averaged probability in % that a specific pattern occurred consecutively at least 3, 4, or 5 times.

5 Discussion

The beauty of this work lies within its simplicity. We combined two simple ideas in order to create a tool that was able to identify most of the typical manipulations committed by inspectors in manufacturing processes. Please note that this tool only points out samples that follow a specific pattern. The appearance of a pattern alone does not yet mean that a manipulation actually took place. In case of an apparent manipulation, an expert should always take a closer look at the corresponding samples and, if necessary, remeasure the affected products. One of the benefits to this tool is an overall increase in efficiency. The effort required to recheck the

suspicious samples recommended by this tool is considerably lower than the effort required to double or triple check all measurements during a manufacturing process. The tool also defuses the worries regarding worker self-inspections stated by Hirsch-Kreinsen in 1997, as it significantly decreases the likelihood of quality data manipulations by providing enhanced monitoring and identification possibilities.

Acknowledgements The authors would like to express their gratitude to CAQ AG Factory Systems in Germany for funding this research, providing information, and establishing contact with experts in quality management.

References

1. Hirsch-Kreinsen, H.: Organisation und Mitarbeiter im TQM, vol. 1. Springer, Heidelberg (1997). http://doi:10.1007/978-3-642-60397-6
2. Sommerhoff, B., Brecht, A., Fiegler, M.: Moderne Ansätze der Qualitätssicherung in der Serienfertigung. Expertenwissen f. DGQ-Mitglieder (2013), Deutsche Gesellschaft f. Qualität e.V. (DGQ), Frankfurt am Main. https://www.dgq.de/z/schnupper/WP_QS.pdf
3. Sommerhoff, B., Brecht, A., Fiegler, M.: Gut gerüstet auf neue Wege. QZ 59, 16–19 (10/2014). www.qz-online.de/882067
4. Brück, D.: Validierung von Messdaten in der Produktion mit mathematischer Optimierung und Machine Learning. Masterarbeit, TU Kaiserslautern (2018)
5. Freund, Y., Shapire, R.E.: A Decision-theoretic generalization of on-line learning and an application to boosting. J. Comput. Syst. Sci. **55**(1), 119–139 (1997). http://doi:10.1006/jcss.1997.1504
6. Smith, T.F., Waterman, M.S.: Identification of common molecular subsequences. J. Mol. Biol. **147**, 195–197 (1981). http://doi:10.1016/0022-2836(81)90087-5
7. Gotoh, O.: An improved algorithm for matching biological sequences. J. Mol. Biol. **162**, 705–708 (1982). http://doi:10.1016/0022-2836(82)90398-9

Modeling the Egg Packing Station Planning Problem

Reginald Dewil, Johan Philips, Jan Jaap Kempenaar, and Dirk Cattrysse

1 Introduction

Eggs are a major European export product. The European Commission forecasted an average monthly production of 7900 million eggs in the European Union in 2018 and with prices hovering around €120 per 100 kg for A-grade eggs, it is an important multi-billion euro industry[1]. In the European Union, eggs are typically produced in laying farms, palletized and sent to a packing station where the eggs are graded by an *egg grader*. Egg graders are large egg sorting machines that grade a continuous supply of incoming eggs into different weight classes, hereafter referred to as *grades*, and remove damaged and otherwise unsuitable eggs (dirty, rotten, bloody). The eggs are subsequently allocated to packing lanes where they are packed in carton boxes destined for retailers. Figure 1 shows an Omnia PX egg grader of machine builder MOBA with 16 packing lanes and 1 end lane.

The weight distribution of the incoming supply of eggs can be approximated by a normal distribution. Since the average weight and standard deviation changes very gradually as the chickens age, the weight distribution can be reliably estimated per egg laying farm based on the weight distribution of the egg production of the previous day. In the packing station, pallets coming from different farms are available at the start of the packing shift, each with their own weight distribution. Put otherwise, with every pallet change, the weight distribution of incoming eggs might change.

R. Dewil (✉) · J. Philips · D. Cattrysse
KU Leuven, Leuven, Belgium
e-mail: reginald.dewil@kuleuven.be

J. J. Kempenaar
MOBA B.V., Barneveld, Netherlands
http://www.moba.net/page/en

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_52

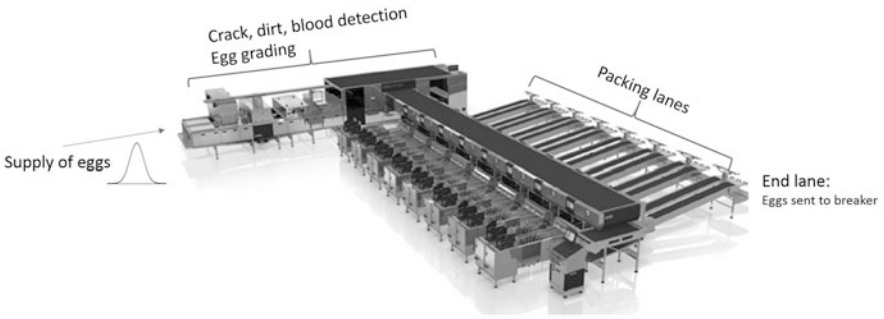
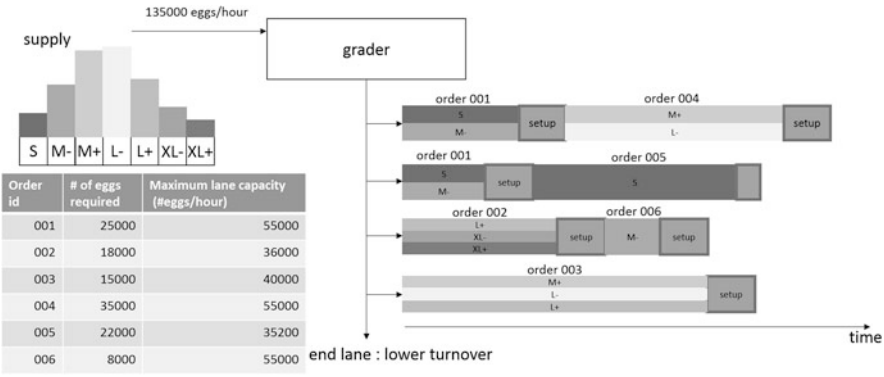


Fig. 1 An Omnia PX egg grader (MOBA B.V.)[2]



Order id	# of eggs required	Maximum lane capacity (#eggs/hour)
001	25000	55000
002	18000	36000
003	15000	40000
004	35000	55000
005	22000	35200
006	8000	55000

Fig. 2 The planning problem

The orders specify the number of eggs required and the allowed grades. The number of eggs required for a given order can vary to a great extent: in the range of 200 eggs up to 160 thousand eggs. The allowed grades can be a single specific grade, such as *extra large* (*XL*) eggs or be composed of multiple grades, such as: *medium*, *large*, and *extra large* (*M, L, XL*). A single order can be fulfilled on multiple packing lanes simultaneously. Since the boxes destined for the same order need to be grouped to be palletized, it is preferable that these so-called sub orders are assigned to adjacent lanes in order to minimize any additional handling. However, this is not a hard constraint.

Figure 2 represents the planning problems in its entirety. The goal is to create sub order—lane allocation such that as many available orders are completely fulfilled given the available egg supply and profit is maximized. This implies minimizing the sequence-dependent setup times and minimizing the number of eggs sent to the end lane. Eggs sent to the end lane are destined for a breaker machine that separates yolk from white which are sold separately at much lower prices than an intact packaged egg. Eggs will be sent to the end lane if there is insufficient packing capacity available to process the egg's grade, or put otherwise, there are insufficient active orders that accept the egg's grade.

Given sub order-lane allocations, the grader will assign individual eggs to active orders that are allowed to pack that egg's grade using a heuristic that tries to assign the eggs equally to the packing lanes. Although some uncertainty remains on how long it will take until a similar grade egg will be graded, the small lane buffers are properly dimensioned to cover any short-term deviation from the normal distribution assumption. Therefore, egg flows resulting from a normally distributed supply can be safely used to determine the number of eggs being allocated to a specific lane. More specifically, Eqs. (1) and (2) approximate the allocation heuristic used by the grader. First, an initial egg flow $f_{g,o}^{initial}$ per grade g to each sub order o is determined using Eq. (1). Let s_g be the egg supply per time unit of grade g and O_g be the set of active sub orders that have grade g as an allowed grade, then the initial egg flow for sub order o of grade g is:

$$f_{g,o}^{initial} = s_g \frac{\frac{1}{\sum_{g' \in o.grades} s_{g'}}}{\sum_{r \in O_g} \frac{1}{\sum_{g'' \in r.grades} s_{g''}}} \tag{1}$$

This initial egg flow is subsequently adjusted if one of these flows would exceed the packing capacity of the lane using Eq. (2).

$$f_{g,o} = f_{g,o}^{initial} \frac{min(o.lane.capacity, \sum_{g \in o.grades} f_{g,o}^{initial})}{\sum_{g \in o.grades} f_{g,o}^{initial}} \tag{2}$$

Consequently, the eggs being sent to the end lane for each grade g is the difference between the sum of the egg flows to each sub order and the total supply of that grade (Eq. (3)).

$$f_{g,endlane} = s_g - \sum_{o \in O_g} f_{g,o} \tag{3}$$

We can conclude from these equations that if anything changes in the active lanes, a breakdown, an end or a start of a sub order on a lane, the egg flows to multiple lanes are affected since the number of active orders of a given grade changes. This has a major impact on the computational complexity of the scheduling problem since the packing lanes cannot be considered as independent machines.

2 Solution Representation and Evaluation

In many basic production scheduling problems or routing problems, in order to result in a Gantt chart as depicted in Fig. 2, a simple two dimensional array of orders suffices. The rows represent the different machines, vehicles, or in this case, the lanes and the elements in the rows represent the sequence of jobs to be executed or stops to be visited. Each row is independent and the objective function value of a

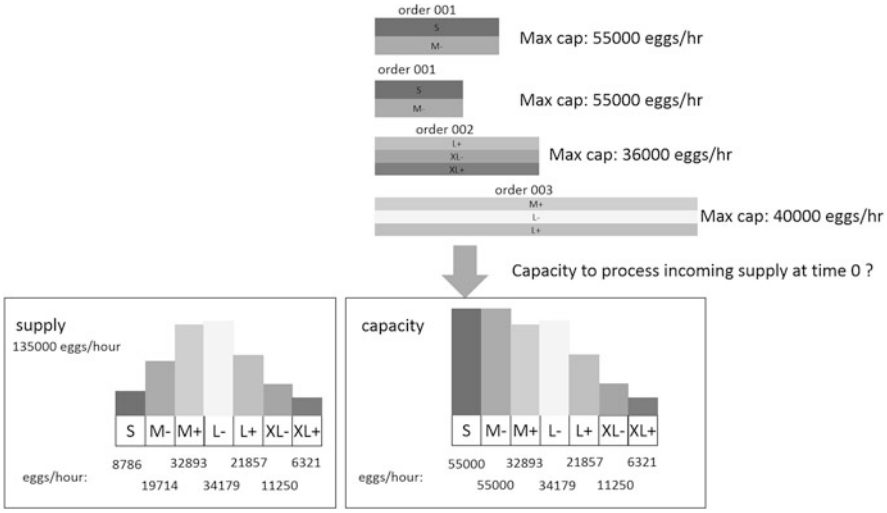


Fig. 3 A sequence of orders and pallets results in a sequence of events each with its own egg flow

complete solution can be determined by adding the objective function values of the individual machine schedules or vehicle routes. This observation lies at the basis of the many successful local search or metaheuristic optimization approaches since a small perturbation can be efficiently evaluated within the route or machine itself and not bother with looping over the complete solution, and thus millions of new solutions can be evaluated in short computation times.

Since in the packing problem, the lane schedules are not independent and the egg flows directly impact the completion times of orders, it is not trivial to identify a local perturbation that can be evaluated in constant time. Whereas in typical scheduling problems, an order start time directly fixes the order end time, in this problem, the end time is dependent on the egg flows. As depicted in Fig. 3, these egg flows change at every event, e.g. *start pallet*, *end pallet*, *start sub order*, *end sub order*, *start setup*, *end setup*.

A single solution can then be evaluated by looping once over the solution and iteratively identifying the next event to occur based on the current egg flow and determining the new egg flow and the number of eggs processed of the pallet and to the sub orders at this event. This is an $O(n+m)$ operation with n being the number of sub orders and m being the number of pallets. Given the efficacy of $O(n^2)$ local move operators such as *remove-reinsert* and *2-opt* in similar routing or scheduling problems and the fact that $O(n^3)$ local move operators are computationally acceptable for the problem sizes of daily instances ($\leq$ 100 orders, $\leq$ 100 pallets, $\leq$ 32 lanes) leads us to conclude that using a classical $O(n^2)$ operator with the event-based $O(n)$ evaluation will most likely lead to an effective optimization approach utilizing $O(n^3)$ local move operators.

A construction heuristic was developed that greedily schedules orders on lanes as they become available. Its greedy criterion is to minimize the eggs being sent to the grader in the resulting egg flow. Ties are broken by selecting the order with the fewest eggs remaining to be fulfilled. In order to generate multiple initial solutions, the construction heuristic was implemented in a Greedy Randomized Adaptive Search Procedure (GRASP)[3]. The GRASP heuristic was run on a real-life benchmark from one of the more efficient egg packing stations in the Netherlands. Comparing to a schedule generated by a manual planner, the GRASP heuristic already improves the daily profit by 6.5%.

In order to determine an upper bound on the achievable profit, the complete problem can also be modeled as a Transportation Problem: the sources are all the incoming eggs and the sinks are the complete set of orders. A source is created for each grade. Assuming a perfect utilization of the lanes and disregarding setups, an upper bound on the profit can be determined. The original schedule and the GRASP schedule showcase gaps of 13% and 7% respectively with this upper bound. This is a major motivation to continue the research not only in adding additional constraints and features but also in searching for better optimization approaches.

2.1 Egg Lane Allocation

The allocation equations imply that the more grades that are permitted by an active order, the less eggs will be assigned to that lane from a specific allowed grade. Furthermore, it implies that, for a specific grade, eggs will be sent to all lanes that are permitted to process this grade. This can actually lead to unwanted effects as can easily be seen through the following example. Consider two orders, each requiring 100 eggs. The first order can be fulfilled by Small or Medium eggs and the second order can only be fulfilled by Small eggs. Suppose that both orders are scheduled on separate lanes with infinite processing capacity. Furthermore, suppose that a supply of 100 Small and 100 Medium eggs is available. It follows that both orders can easily be fulfilled to completion. Assume that the grader speed is 100 eggs per second (eps). Therefore, there will be a supply of 50 eps of Medium eggs and 50 eps of Small eggs. However, the above defined egg allocation equations result in the following egg flows per second: $f_{S,1} = 16.67$, $f_{M,1} = 50$, $f_{S,2} = 33.33$. This means that there is an egg flow of 66.67 eggs per second to the first order which will be fulfilled after 1.5 s after which a supply of 25 Medium eggs remains which is not allowed to be allocated to the second order.

This simple example illustrates a problem that effectively occurs in practice. However, this problem is, on the one hand, exacerbated by lane capacity constraints which result in eggs being sent to the end lane and, on the other hand, alleviated by reallocating eggs to downstream lanes if no order-specific text is printed on them.

From an Operations Research perspective, this allocation problem is clearly a Transportation Problem and initial experiments suggest that replacing the machine's heuristic by an optimal algorithm that solves the Transportation Problem can improve daily profitability by an additional 2%.

3 Discussion and Outlook

The egg planning problem as presented above is a basic version of the planning problem as it occurs in practice. However, the results we have obtained using a very simple GRASP heuristic is sufficiently promising to continue this line of research. Furthermore, additional improvement potential is identified in improving the egg allocation heuristic and there remains improvement potential by implementing a local search phase in the optimization process.

In order to design a schedule that will be accepted by practitioners on the shop floor, many additional constraints need to be considered and the approach presented above is just a first step. In particular, the above approach does not place a limit on the number of setups that can be executed in parallel but in reality this is limited by the number of people executing the setups.

The packing lanes are subject to frequent short breakdowns which have a major impact on the egg flows. Consequently, the schedules should be constructed such that sufficient capacity for every grade is available at all times to cover one or multiple simultaneous breakdowns.

Sometimes, a single worker operates two lanes simultaneously or multiple packing lanes are linked together downstream by a packing robot. In both cases, the lane capacities are limited by the downstream robot or worker capacity. However, this capacity is shared across the lanes and introduces an extra degree of complexity in the egg allocation and consequently the order scheduling problem.

In the current version, we have assumed that the orders and pallets were given beforehand. However, this problem occurs daily and some orders can be postponed and, similarly, some egg supplies can be processed a day or two later. Hence, we should actually consider a multi-day planning problem.

The end goal is to develop an advanced planning module that can schedule the daily operations and reschedule quickly when unforseen events occur. Furthermore, such a module can be used to determine the optimal machine configuration for new packing stations supporting the commercial process, i.e. the type of grader, the number of lanes, using downstream packing robots, etc.

References

1. European Commission: Egg Market Situation (June 1st, 2018). https://circabc.europa.eu/sd/a/18f7766e-e9a9-46a4-bbec-94d4c181183f/17.05.2018_eggs.pdf
2. MOBA. http://www.moba.net/page/en/
3. Feo, T.A., Resende, M.G.: Greedy randomized adaptive search procedures. J. Glob. Optim. **6**(2), 109–133 (1995)

A Multi-site Facility Layout and Product-to-Site Assignment Problem

Bernd Hillebrand

1 Introduction

Job shop production systems are characterized in general by a heterogeneous and complex material flow between production units, which perform different production operations. This is caused by the interdependencies of producing different products by means of the same production units according to different work schedules. Facility layout planning being based on such a material flow requires a compromise in placing the respective production units on the floor: The minimum total costs of material flows in the system can only be achieved by deviating from product-specific cost minimal material flows (heterogeneity costs).

In a company with more than one production site at its disposal, the sites can focus on smaller disjunctive product programs. In this context the decision field of facility layout planning is extended by two questions: (1) Which products are to be produced at which site, and (2) for which production unit redundancies are necessary? The first question enables the company to disentangle the material flows. That is, to group similar products together in order to achieve more homogeneous material flows at each site. The second question arises, since not only similar products are produced with the same production units. Hence, the effect of disentanglement can be increased by building up redundancies across the sites for production units required by products with heterogeneous material flow. That means, placing production units of the same type to different sites, enlarges the possibilities to achieve homogeneous material flows at the respective sites. In this extended situation the task of layout planning is to balance the trade-off between costs of heterogeneity and redundancy.

B. Hillebrand (✉)
Chair of Production Management and Logistics, TU Dortmund University, Dortmund, Germany
e-mail: bernd.hillebrand@tu-dortmund.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_53

2 A Multi-site Facility Layout and Product-to-Site Assignment Problem

2.1 Problem Description and Modelling Assumptions

Consider a company with S production sites that has to satisfy the demand of a production program of P products. To produce a product $p \in P$ a specific subset of the set of production units I is necessary at a specific site $s \in S$. These production units represent for example machines, workshops and similar units, which are necessary to produce the respective product. The satisfaction of the given demand of p induces a material flow λ_{pij} between production units of type $i, j \in I$ by producing the necessary quantity of this product. It is assumed that the capacity of a production unit is sufficiently large such that every product at a site can be produced by it, if it is necessary. Further at maximum a production unit of type i is allowed at every site. Every site $s \in S$ has a set of locations K_s. Every location $k \in K_s$, for $s \in S$, is able to accommodate every production unit $i \in I$ but not more than one in the given planning horizon. A location k belongs only to one site s. Between the different locations within one site a distance d_{skl}, for $k, l \in K_s$ exists. A transport of the semi-finished products is only possible at one production site such that there is no material flow between different sites. Two kinds of costs are considered in this model: On the one hand costs for transporting semi-finished products between the locations k and l of one site s, which depend on the distance between these locations, the material flow of the assigned objects i and j at the respective locations and the cost factor c^t_{skl}. On the other hand costs for providing a production unit at a site, where the cost factor C^a_{sik} depends on the production unit i, the location k and the site s. Two kinds of decision questions arise in this problem: (1) Should production unit of type i be placed on location k at site s, and (2) should product p be produced at site s? The first question is represented by the variable u_{sik}, which is equal to one, if i is assigned to k at s and otherwise zero. The second question is represented by q_{sp} that indicates the share of the demand of product p that is produced in s. From these two variables the variable v_{skl} can be derived, which describes the material flow between locations k and l at site s.

The objective is to minimize the total costs consisting of provision of production units and transportation costs induced by the material flow.

2.2 Differentiation to Similar Approaches

While this approach is based on a discrete facility layout model, there are some similar approaches, that deal with facility layout planning and multi-site-allocation, altering the product program of a facility and/or building up redundancies of production units. For a deeper literature review regarding facility layout problems, please see for example [1, 2]. In the following, three kinds of similar approaches

are investigated: (1) Multi-floor facility layout problems (MFFLP) deal with facility layout planning problems for a building with more than one floor. Normally the floors are connected by stairs or elevators at respective places. So the floors are similar to the sites in this approach but they are connected by an additional (vertical) distance. The approach of this contribution neglects such connections but instead the possibility of building up redundancies of production units across all sites is possible. MFFLP have only a predetermined number of different production units (in general this number is equal to one), such that these approaches can be handled like single-floor facility layout problems with non-proportional transportation costs regarding the distance because a product-to-site assignment does not take place. For a review regarding MFFLP please see [3]. (2) Integrated machine allocation and layout problems deal with the allocation of more than one kind of production unit. They also consider the capacity of the units for producing different products and describe the possibility to build up redundant machines at a site to shorten the distances between different kinds of machines. While the product program at the respective site is given for such approaches, our contribution deals with redundancies across different sites and a product-to-site assignment. For examples of this class of layout problems, see [4, 5]. (3) Integrated make-or-buy and facility layout problems deal with the possibility to decide over the product program at one site. It is possible to leave the production of some products to an external firm with given costs for outsourcing. In our approach such an outside option is not available and instead of looking only at one site, all products have to be allocated to the given sites. For an example of a layout problem with integrated make-or-buy decision, please see [6].

2.3 Model Statement

The following optimization model seeks to determine the optimal solution of the multi-site facility layout and product-to-site assignment problem:

$$\min Z = \sum_{s\in S}\sum_{i\in I}\sum_{k\in K_s} C^a_{sik} \cdot u_{sik} + \sum_{s\in S}\sum_{k\in K_s}\sum_{l\in K_s} c^t_{skl} \cdot d_{skl} \cdot v_{skl} \tag{1}$$

$$\sum_{k\in K_s} u_{sik} \leq 1 \quad \forall s \in S; i \in I \tag{2}$$

$$\sum_{i\in I} u_{sik} \leq 1 \quad \forall s \in S; k \in K_s \tag{3}$$

$$\sum_{s \in S} q_{sp} \geq 1 \quad \forall p \in P \tag{4}$$

$$v_{skl} \geq \sum_{p \in P} q_{sp} \cdot \lambda_{pij} - M_{ij} \cdot \left(2 - u_{sik} - u_{sjl}\right) \quad \forall s \in S; i, j \in I; k, l \in K_s \tag{5}$$

$$\sum_{p \in P} \left[q_{sp} \cdot \sum_{j \in J} \left(\lambda_{pij} + \lambda_{pji}\right) \right] \leq \sum_{j \in J} \left(M_{ij} + M_{ji}\right) \cdot \sum_{k \in K_s} u_{sik} \quad \forall s \in S; i \in I \tag{6}$$

$$v_{skl} \geq 0 \quad \forall s \in S; k, l \in K_s \tag{7}$$

$$u_{sik} \in \{0; 1\} \quad \forall s \in S; i \in I; k \in K_s \tag{8}$$

$$0 \leq q_{sp} \leq 1 \quad \forall s \in S; p \in P, \tag{9}$$

where M_{ij} is a very large number and $M_{ij} = \sum_{p \in P} \lambda_{pij}$ is suggested.

The problem (1)–(9) is based on quadratic assignment problem (QAP) [7] and uses the linearization-approach developed by [6] for the integrated make-or-buy and facility layout problem. The objective function (1) consists of two parts, where the first part represents the costs of providing production units at the respective locations at the different sites, where the second part represents transportation costs of the induced material flow. Constraint (2) ensures that every production unit is provided not more than one time at a facility, while (3) ensures that no location accommodates more than one production unit at neither site. Constraint (4) guarantees that every product is assigned to at least one site. Constraint (5) assigns the material flow between two production units to the locations, where they are placed. Constraint (6) ensures that a production unit is provided at a site where the production of a product takes place, which needs this unit for production (indicated by a material flow from or to this object). Constraints (7)–(9) define the feasible range of the decision variables.[1]

[1] It can be shown that this problem has an optimum solution, where the problem faces the additional restriction that the variable q_{sp} is binary, such that the whole demand of a product is satisfied by the production of one site. For details see [6].

3 Numerical Tests

For our numerical tests we assume that every site has six locations and that there are six kinds of production units. The distance between the locations that are placed in a row amounts to one distance unit per move to a neighboring location. Other arrangements, like two rows with three locations each, lead to similar results, but the effect is lower due to a better connectivity among the different locations to each other.

We assume products either use all six kinds of machines or use only three out of six for production. From every kind of production unit in use the whole material is sent at maximum to only one other production unit and every unit is only used once for a product, so that the material flow can be represented in a graph as a tree, where the nodes are the machines in use. If you neglect the direction of the material flow, this will lead to 1296 products that use all machines and to 60 products that use three out of six. Neglecting the material flow direction is possible, because of a symmetric distance. Every material flow connection is set to one unit.

Two indicators try to measure the homogeneity: ϑ^1 stands for the fraction of material flow connections in use by the production program at one site and ϑ^2 counts the number of material flow connections which are used by more than one product at one site divided by ϑ^1.

The first part of this study investigates how homogeneity of the material flow affects the reduction of heterogeneity costs. To this end four different testing settings are investigated (I: six products/six stages; II: ten products/six stages; III: six products/three stages and IV: ten products/three stages). In every test setting the products were chosen randomly 20 times with equal probability and solved each for one and for two sites. To compare the instances, the transportation cost rate is set to one, while the provision costs are zero.

By comparing the numerical tests with respect to ϑ^1 and ϑ^2 (see Fig. 1), we investigate that the possibility of having two sites reduces the average costs of the material flow connections. The reduction of different material flows seems to be the reason, because the flow becomes more homogenous at one site and the heterogeneity costs sink (top diagrams in Fig. 1). On the other hand ϑ^2 seems to have a lower indication to the mean cost reduction (bottom diagrams of Fig. 1) and is outpowered by ϑ^1.

The second part investigates the trade-off between costs of heterogeneity and redundancy across different sites. To this end, C^a_{sik} is altered systematically for the testing settings III and IV to investigate production sites, which do not require all kinds of machines for production. By altering the provision costs and having two production sites, we can see that the higher the provision costs are, the more likely one site becomes ‘generalist’ while the other one becomes a ‘specialist’. Transport flow costs decrease at the specialist, because it focuses on a smaller production program, where it needs less production units, while the generalist produces the other products (see left diagram of Fig. 2). This leads to a more homogenous product program at the specialist, while the generalist faces a more complex material flow

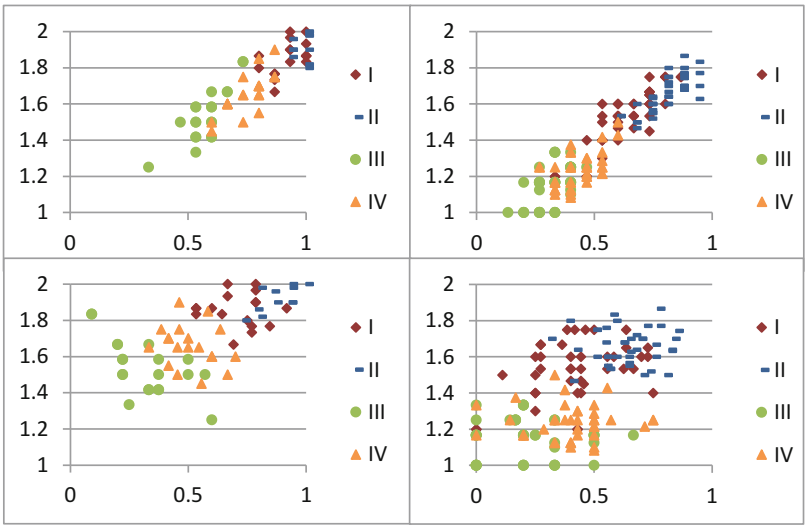


Fig. 1 Comparison between the average cost of a material connection (y-axis) and ϑ^1 at one site (x-axis at the top diagrams) or ϑ^2 at one site (x-axis at the bottom diagrams) of testing settings I–IV (20 instances each) with one site (left) and two sites (right)

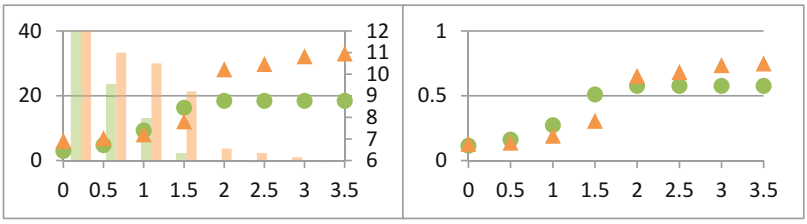


Fig. 2 Comparison between the average absolute transport cost difference of the respective two sites (marking points left diagram and left y-axis) average number of machines in use over both sites (columns left diagram and right y-axis) and average difference in ϑ^1 of the two respective sites (right diagram) while altering the provision costs of machines (x-axis) of scenarios III and IV (colors are the same as in Fig. 1)

(see right diagram of Fig. 2). With more sites on hand, it is likely that these sites are specialists, while the material flow of the generalist also becomes more and more one of a specialist.

4 Conclusion

This contribution provides a new integrated approach to deal with the product-to-site assignment while solving the facility layout problem at every site. A numerical study investigates how homogeneity of the material flow reduces the transportation costs,

while provision costs of production units lead to differently equipped production sites. By extending the numerical study, future research might be able to find more adjusting parameters of this problem to give ex ante advices for assigning different products with a given production structure to one site.

References

1. Drira, A., Pierreval, H., Hajri-Gabouj, S.: Facility layout problems: a survey. Annu. Rev. Control. **31**, 255–267 (2007)
2. Gössinger, R., Hillebrand, B.: Layoutplanung. In: Corsten, H., Gössinger, R., Spengler, T.S. (eds.) Handbuch Produktions- und Logistikmanagement in Wertschöpfungsnetzwerken, pp. 553–591. De Gruyter Oldenburg, Berlin/Boston (2018)
3. Ahmadi, A., Pishvaee, M.S., Akbari Jokar, M.R.: A survey on multi-floor facility layout problems. Comput. Ind. Eng. **107**, 158–170 (2017)
4. Urban, T.L., Chiang, W.-C., Russell, R.A.: The integrated machine allocation and layout problem. Int. J. Prod. Res. **38**(13), 2911–2930 (2000)
5. Zhao, Y., Wallace, S.W.: Appraising redundancy in facility layout. Int. J. Prod. Res. **54**(3), 665–679 (2016)
6. Hillebrand, B., Pishchulov, G.: Integrated make-or-buy and facility layout problem with multiple products. In: Mattfeld, D.C., Spengler, T.S., Brinkmann, J., Grunewald, M., et al. (eds.) Logistics Management, pp. 121–132. Springer, Cham (2016)
7. Koopmans, T.C., Beckmann, M.: Assignment problems and the location of economic activities. Econometrica. **25**(1), 53–76 (1957)

A Scatter Search Approach for the Facility Layout Problem with Aisle Design

Armin Klausnitzer

1 Introduction

The designer of a plant layout is tasked with the efficient arrangement of manufacturing and service facilities to minimize travel distance. This problem is referred to as the facility layout problem (FLP) and consists of (1) shaping and arranging facilities within the layout, (2) locating material input and output (I/O) points, and (3) designing the material handling system. Due to the complexity, these related problems are traditionally solved sequentially [4]. Since the location of the aisles is not known in the early design phases, most approaches solve the first and second step using rectilinear distances. Due to bypass blocking facilities, the distances can increase significantly when applied to real-world situations [1]. Moreover, between 5 and 40% of the floor space is occupied by aisles [10, 11]. Therefore, the literature suggests boosting the areas of facilities before step 1 to obtain feasible arrangements in a given plant area when later implementing aisles [8]. However, estimating a boosting coefficient is crucial, as a small value does not provide enough space to plan aisles, and a value that is too large leads to less layout density. Consequently, sequentially planning for these dependent problems can lead to poor solutions and thereby result in costly replanning.

Apart from the traditional approaches that use rectilinear distances, only a few integrate the path design, e.g. [1, 4]. However, there is no approach that solves all three planing tasks while implementing aisle areas along the paths. This study aims to close this gap by proposing a linear programming and graph-based scatter search approach. The remainder of the paper is organized as follows: Sect. 2 describes the layout generation method and the scatter search; Sect. 3 presents the results of a

A. Klausnitzer (✉)
TU Dresden, Dresden, Germany
e-mail: armin.klausnitzer@tu-dresden.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_54

computational study; and Sect. 4 concludes the paper with findings and suggestions for future research.

2 Solution Approach

2.1 Storage and Calculation of Layouts

Our approach to constructing a layout solution requires four main steps. Based on Kim and Goetschalckx [4], a feasible layout is constructed first, and the travel paths along the contours of the facilities are identified second. In the third step, a clearance along the travel paths applying the minimum aisle widths is implemented. Finally, the objective function value is evaluated.

Step 1: Generating a Preliminary Layout: For the first step, a sequence-pair representation consisting of two facility sequences, Γ^+ and Γ^- is applied to encode a possible layout solution. This coding is known to be efficient since each permutation on Γ^+ and Γ^- guarantees a feasible layout with respect to the avoidance of the overlapping of facilities [7]. Using a given sequence pair $(\Gamma^+|\Gamma^-)$, the following linear program (LP) can be solved.

Parameters

i, j	Facility indices $i, j = 1, \ldots, N$, where N is the total number of facilities
f_{ij}	Material flow between facilities i and j, where $f_{ij} \geq 0$
L^s	Side length of the floor space in direction $s \in \{x, y\}$, assuming a $(0, 0)$ origin
a_i	Area requirement for facility i
lb_i^s, ub_i^s	Lower and upper limits on the half of side length l_i^s
$\bar{x}_{i\lambda}$	Tangential support of facility i in discretization point $\lambda = 0, \ldots, \Delta - 1$
α_i	Maximum permissible ratio between the longest and shortest side of facility i
l_{min}	Minimum side length of facility i

Decision Variables

d_{ij}	Distance between facility i and j
c_i^s	Centroid point of facility i with respect to the direction s
l_i^s	Half of side length of facility i in direction s, where $2l_i^s \geq l_{min}$
IO_i^s	Location of the I/O point of facility i with respect to the direction s

$$\min \sum_{i=1}^{N} \sum_{\substack{j=1,\\ j\neq i}}^{N} f_{ij} d_{ij} \tag{1}$$

Subject to:

$$d_{ij} = \left| IO_i^x - IO_j^x \right| + \left| IO_i^y - IO_j^y \right| \quad \forall i \neq j \tag{2}$$

$$c_i^s - l_i^s \leq IO_i^s \leq c_i^s + l_i^s \quad \forall i, s \tag{3}$$

$$l_i^s \leq c_i^s \leq L^s - l_i^s \quad \forall i, s \tag{4}$$

$$lb_i^s \leq l_i^s \leq ub_i^s \quad \forall i, s \tag{5}$$

$$a_i l_i^x + 4\bar{x}_{i\lambda}^2 l_i^y \geq 2a_i \bar{x}_{i\lambda} \quad \forall lb_i^x \leq \bar{x}_{i\lambda} \leq ub_i^x \tag{6}$$

$$c_i^x + l_i^x \leq c_j^x - l_j^x \quad \forall i \neq j, i \text{ precedes } j \text{ in } \Gamma^+ \text{ and } \Gamma^- \tag{7}$$

$$c_i^y + l_i^y \leq c_j^y - l_j^y \quad \forall i \neq j, j \text{ precedes } i \text{ in } \Gamma^+, i \text{ precedes } j \text{ in } \Gamma^- \tag{8}$$

The objective function (1) minimizes the sum of the products of material flow intensities and distances. The rectilinear distances (2) can be linearized using $IO_i^s - IO_j^s \leq d_{ij}$ and $IO_j^s - IO_i^s \leq d_{ij}$. Constraints (3) and (4) ensure that I/O points must be located within the respective facility, and each facility is forced to be located within the floor boundaries, respectively. Lower and upper limits restrict the side length of each facility (5) using $lb_i^s = a_i/4ub_i^s$ and $ub_i^s = \min\left\{\sqrt{a_i \alpha_i}, \max_s \{L^s\}\right\}/2$ [9]. To maintain a given area, the exact hyperbolic area constraint $a_i = 4l_i^x l_i^y$ can be linearized using a polyhedral outer-approximation (6) developed by Sherali et al. [9]. Therefore, the side lengths of departments are constrained using Δ discretization points, which tangentially support curve $l_i^y = a_i/4l_i^x$. The values of the tangential support points $\lambda = 0, 1, \ldots, \Delta - 1$ for each facility i equal $\bar{x}_{i\lambda} = lb_i^x + \frac{\lambda}{\Delta - 1}(ub_i^x - lb_i^x)$. According to the sequence pair $(\Gamma^+|\Gamma^-)$, the relative locations of facilities are determined and not allowed to overlap using constraints (7) and (8).

Step 2: Improving I/O Points: Since the I/O points of the LP are optimized regarding the rectilinear distance metric, the I/O points with respect to the path-based distance measure can differ. Therefore, the layout solution is transferred to an undirected graph $G = (V, E)$. Vertices V are points at which horizontal and vertical boundaries of facilities meet, as well as points of their mirror images at neighboring facilities. These points are connected by a set of horizontal and vertical edges E. To obtain an efficient set $VE \in V$ of dominant I/O candidates, Kim and Kim [5] suggest forming the smallest rectangle that borders at least one vertex of each department. Subsequently, the distances of the paths between each pair of I/O candidates, where $f_{ij} > 0$, are calculated using Dijkstra's algorithm.

To determine the I/O points, the facilities i and j are sorted in descendent order of their flow intensity, i.e. $i, j \leftarrow argmax_{i,j \in 1 \ldots N} \left\{f_{ij}\right\}$. Using this order of facilities, their I/O points p_i^{IO}, q_j^{IO} are determined according to the lowest distance, i.e. $p_i^{IO}, q_j^{IO} \leftarrow argmin_{p_i, q_j \in VE} \left\{d_{p_i q_j}\right\}$. Without changing defined I/O points, the allocation repeats until all facilities are assigned one I/O point [1].

Step 3: Allocating Aisle Area: After the I/O points are determined, the facilities along the paths are identified. If a path passes between two neighboring facilities i and j, constraints (9) and (10) replace (7) and (8). This ensures a clearance of aisle width a^w along the travel paths.

$$c_i^x + l_i^x + a^w \leq c_j^x - l_j^x \quad \forall i \neq j, i \text{ precedes } j \text{ in } \Gamma^+ \text{ and } \Gamma^- \tag{9}$$

$$c_i^y + l_i^y + a^w \leq c_j^y - l_j^y \quad \forall i \neq j, j \text{ precedes } i \text{ in } \Gamma^+, i \text{ precedes } j \text{ in } \Gamma^- \tag{10}$$

If a path passes between a facility and the left or lower boundary of the floor space, constraint (4) is replaced by constraint (11) with $s = x$ and $s = y$, respectively. Similarly, constraint (12) with $s = x$ and $s = y$ is used if the path passes between a facility and the right or upper boundary.

$$l_i^s + a^w \leq c_i^s \leq L^s - l_i^s \quad \forall i, s \tag{11}$$

$$l_i^s \leq c_i^s \leq L^s - l_i^s - a^w \quad \forall i, s \tag{12}$$

Repeating Step 1 with constraints (9) to (12) results in a facility arrangement that considers aisle areas along the paths.

Step 4: Adjusting I/O Points and Layout Evaluation: Since the coordinates of facilities can change in Step 3, the I/O points must be adjusted. Therefore, Step 2 repeats searching for the shortest paths and I/O points. To ensure that the new I/O points are located near the aisles defined in Step 3, the candidate I/O points of each facility i are limited to vertices $VN_i \in V$ representing the relative locations of the respective original I/O points.

Finally, the algorithm checks for aisle areas along the paths and calculates the objective function value (OFV) in (1) using the distances d_{ij} of Dijkstra's algorithm or returns to Step 3, if necessary.

2.2 Scatter Search Algorithm

Every layout is encoded in a pair of facility sequences $(\Gamma^+ \mid \Gamma^-)$. To search for a layout solution with a superior OFV, a scatter search algorithm changes the order of facilities in $(\Gamma^+ \mid \Gamma^-)$ and consequently the corresponding layout solutions using the algorithm shown in Sect. 2.1. Based on a template by Glover [2] and the heuristic approach of Kothari and Gosh [6], the algorithm comprises the following five methods:

1. The *diversification generation method* builds a large set U of diverse permutations by randomly interchanging facilities in each sequence of pair $(\Gamma^+ \mid \Gamma^-)$, starting from a random initial solution. Then, the algorithm chooses an elite set

$ES \in U$ showing the best OFVs to focus on superior solutions. Finally, the initial population $P \in ES$ is generated by choosing permutations of ES according to their coding distances. The distance is calculated by the sum of the positional distances of each facility in two permutations. For example, facility 1 in Γ^+ of permutations (1 2 3 | 2 1 3) and (2 3 1 | 3 1 2) is located on the first and third position and shows a distance of $|1-3| = 2$. The total coding distance for this example is $|1-3|+|2-1|+|3-2|+|1-3|+|2-2|+|3-1| = 8$. The first two permutations in population P will be those with the largest coding distance in ES. Then, the algorithm iteratively adds the permutations with the largest distance to both of the previously added permutations until $|P|$ permutations are chosen.

2. The *improvement method* attempts to improve each permutation in the initial population P by randomly changing the position of one facility in either Γ^+ or Γ^-. The new permutation replaces the original one in population P if its OFV is superior.
3. Subsequently, the *reference set update method* generates and maintains a reference set B of B_1 good and B_2 diverse permutations. Therefore, the population P is sorted in increasing order of their OFVs. The first B_1 and last B_2 permutations of the sorted P become members of B.
4. The *subset generation method* produces $\binom{|B|}{|\widehat{B}|}$ different subsets $\widehat{B}$ of the reference set B as a basis for creating new permutations through combinations in the following method.
5. Finally, for each subset $\widehat{B}$, the *solution combination method* combines the permutations using the partially-mapped crossover operator (PMX) [3]. The PMX applies to the entire permutation of $(\Gamma^+ \mid \Gamma^-)$ while ensuring N different facilities in Γ^+ and Γ^-. After applying the improvement method (method 2) to each new permutation, the new permutations replace those with the worst OFVs in the reference set B if their OFVs are superior. Otherwise, the new permutation is discarded.

The algorithm repeats methods (4) and (5) until no new permutation enters the reference set B. The algorithm terminates after reporting the layout solution resulting from the permutation with the best OFV in the reference set B.

3 Computational Results

Since this integrated problem has not previously been solved, no comparable solutions exist. Therefore, the approach is evaluated using the best-known layout solutions with integrated path design reported in [1], in which we included the aisle area along the travel paths in a sequential way. In detail, we applied the relative positions of facilities of the reported best layout solutions to solve the model of Step 1 and adopt the reported relative locations of I/O points. After calculating the shortest paths using Dijkstra's algorithm, we applied Step 3 to insert aisle areas

Table 1 Numerical results

	Shape restriction		Seq. method	Sim. method			Improvement
Data set	α	l_{min}	OFV^{seq}	OFV_{best}	$\overline{OFV}$	s_{OFV}	[%]
vC10	–	5	6860.89	5394.86	7116.14	931.13	21.37
Ba12	–	1	Infeasible	4831.21	5561.12	348.18	100.00
Ba14	–	1	3117.42	2623.13	3784.47	438.42	15.86
AB20	4	1	1699.77	1529.63	–[a]	–[a]	10.01

[a]Only 2 out of 10 runs led to feasible solutions

along the paths and calculated the OFV. For our experimental setup, we assume a boosting coefficient of 25% and calculate the net areas of facilities i using $a_i \longleftarrow \frac{a_i}{1.25}$. To test a similar proportion of aisle width and planning area within each test instance, aisle widths are assumed to be $a^w = 0.08\sqrt{L^x L^y}$. After a pilot study, the parameters $\Delta = 20$, $|U| = 1000$, $|E| = 400$, $|P| = 100$, $|B| = 15$, $B_1 = 10$, $B_2 = 5$, and $|\widehat{B}| = 2$ were determined and each test was repeated 10 times. All tests were conducted on a computer with an Intel Core i7 with 2.2 GHz and 16 GB of RAM. The algorithm was coded in C++ using the graph library LEMON and CPLEX v.12.7.1. Table 1 shows the OFV of the sequential design (OFV^{seq}) and the best OFV (OFV_{best}), the mean OFV ($\overline{OFV}$), and the standard deviation (s_{OFV}) using our approach as well as the difference between OFV^{seq} and OFV_{best}. Illustrations of the best layouts are available on request.

The study shows that for all problems the subsequent integration of aisle areas leads either to infeasible or inferior solutions in comparison to a simultaneous design. A simultaneous design efficiently arranges facilities and aisle areas.

4 Conclusions

A large amount of research regarding the facility layout problem has been conducted to improve single planning steps of the sequential design problem. We proposed an approach that considers the arrangement of facilities, the location of I/O points and the aisle design by minimizing the total travel distances along the travel paths. Therefore, a scatter search approach modifies a sequence pair encoding one layout solution. Based on this coding, a linear program to build an initial layout and a graph to find the shortest paths between the facilities are used. Then, the locations of paths are determined and the layout is adjusted to implement aisle areas. The computational results show that integrating aisle areas within the arrangement problem results in considerably better layout solutions and enables layouts without a boosting coefficient that is hard to estimate. Therefore, the layout designer is forced to repeat costly replanning steps less frequently.

The algorithm uses a greedy I/O location method and is limited to a static problem. Future research should focus on the performance of the algorithm such as evaluating other I/O location procedures and on its applicability to a dynamic environment.

References

1. Friedrich, C., Klausnitzer, A., Lasch, R.: Integrated slicing tree approach for solving the facility layout problem with input and output locations based on contour distance. Eur. J. Oper. Res. **270**, 837–851 (2018)
2. Glover, F.: A template for scatter search and path relinking. In: Hao, J.-K., Lutton, E., Ronald, E., Schoenauer, M., Snyers, D. (eds.) Artificial Evolution. Lecture Notes in Computer Science, vol. 1363, pp. 1–51. Springer, Berlin (1998)
3. Goldberg, D., Lingle, R.: Alleles, loci and the traveling salesman problem. In: Grefenstette, J. (ed.) Proceedings of the 1st International Conference on Genetic Algorithms and Their Applications, pp. 154–159. L. Erlbaum, Hillsdale (1985)
4. Kim, J.-G., Goetschalckx, M.: An integrated approach for the concurrent determination of the block layout and the input and output point locations based on the contour distance. Int. J. Prod. Res. **43**, 2027–2047 (2005)
5. Kim, J.-G., Kim, Y.-D.: A branch and bound algorithm for locating input and output points of departments on the block layout. J. Oper. Res. Soc. **50**, 517–525 (1999)
6. Kothari, R., Gosh, D.: A scatter search algorithm for the single row facility layout problem. J. Heuristics **20**, 125–142 (2014)
7. Murata, H., Fujiyoshi, K., Nakatake, S., Kajitani, Y.: Rectangle-packing-based module placement. In: Proceedings of IEEE International Conference on Computer Aided Design, pp. 472–479. IEEE Computer Society, Washington (1995)
8. Scholz, D., Jaehn, F., Junker, A.: Extensions to STaTS for practical applications of the facility layout problem. Eur. J. Oper. Res. **204**, 463–472 (2010)
9. Sherali, H., Fraticelli, B., Meller, R.: Enhanced model formulations for optimal facility layout. Oper. Res. **51**, 629–644 (2003)
10. Sule, D.: Manufacturing Facilities. CRC Press, Boca Raton (2009)
11. Tompkins, J., White, J., Bozer, Y., Tanchoco, J.: Facilities Planning. Wiley, Hoboken (2010)

Part XIV
Project Management and Scheduling

Real-World Staff Rostering via Branch-and-Price in a Declarative Framework

Marco Bender, Sebastian Berckey, Michael Elberfeld, and Jörg Herbers

1 Introduction

Staff scheduling aims at deploying the right amount of staff with the right qualifications at the right time to the right place. A core problem in this domain is the generation of a *staff roster*, i.e., an assignment of an *activity* to each employee and day. An exemplary roster for a service station with three employees is shown below: an activity is given by a shift (EARLY, LATE, or DAY-OFF) and a function (*repair* or *phone*).

	July 1	July 2	July 3	···
Peter	EARLY	EARLY	LATE	
	Phone	*Repair*	*Repair*	
Laura	LATE	DAY-OFF	EARLY	
	Repair	–	*Repair*	
Paul	EARLY	EARLY	LATE	
	Repair	*Phone*	*Phone*	

In its simplest version, *roster generation* asks for creating a roster for a certain planning period that meets resource requirements with the available workforce. In practice, constraints and objectives related to employee needs like preferred shifts and rest times, company goals like productivity, and customer goals like high service levels need to be taken into account. Some constraints need to be strictly obeyed while others may be treated as soft rules. The number of constraints as well as the

M. Bender (✉) · S. Berckey · M. Elberfeld · J. Herbers
INFORM GmbH, Workforce Management Division, Aachen, Germany
e-mail: marco.bender@inform-software.com; sebastian.berckey@inform-software.com; michael.elberfeld@inform-software.com; joerg.herbers@inform-software.com

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_55

often contradicting objectives make this a complex task. Planning periods that are usually in the range of a month and covering groups of hundreds of employees call for an efficient optimization approach. We refer to [4] for a survey on scheduling and rostering problems.

In the present paper, we outline INFORM's approach towards generating optimized staff rosters. Our optimization algorithm is used to solve large-scale real-world roster generation instances. It is actively being used in aviation (for example, by airlines and ground handlers) and other industries. The problem is modeled as a mixed integer linear program. Due to the exponential number of variables, it is solved using a branch-and-price approach, where the pricing is modeled as a resource-constrained shortest path problem.

In order to quickly adapt the optimizer for new customers, we developed a domain-specific language that allows to easily state the specific business rules of a customer in a declarative fashion. In particular, the mapping from a customer's data to the input of the optimizer and the post-processing of the optimizer results are defined by a customer-specific model in this language.

Organization of This Paper Section 2 defines roster generation and highlights conceptual ideas for its solution in practice. Moreover, results on the performance of our algorithm are presented. Section 3 describes how the optimizer is adapted to requirements of customers via a declarative programming framework.

2 The Roster Generation Problem, Our Solution, and Its Evaluation

In the present section, we first define roster generation in terms of the mixed integer program (MIP) formulation we use. Next, we outline the approach of our optimization algorithm and evaluate its performance on nurse rostering data.

Problem Statement Following [5], we formulate the *roster generation problem* as a MIP: Let E be the set of employees and T be the discretized planning period (like, for example, the days of a month). To each employee $e \in E$, we associate a set R_e of feasible rosters. A roster $r \in R_e$ for employee e consists of exactly one activity $a_{r,t} \in A$ for each slot $t \in T$, where we use the notation A for the set of all possible activities. The set of demands is given by D, and we have an integer target level n_d for each demand $d \in D$. The set of all activities that cover demand $d \in D$ for slot $t \in T$ is denoted by $A_{d,t}$. This allows to formulate the rostering problem as the following MIP:

For every $e \in E$, we use the *roster constraint*

$$\sum_{r \in R_e} x_{e,r} = 1$$

to ensure that a valid solution chooses exactly one roster for employee e. Here, the binary variable $x_{e,r}$ attains the value 1 if employee $e \in E$ is assigned to roster $r \in R_e$.

For every $d \in D$, we use the *demand constraint*

$$\sum_{\substack{e \in E, r \in R_e, t \in T: \\ a_{r,t} \in A_{d,t}}} x_{e,r} + u_d - o_d = n_d$$

to ensure that demand d is met. Possible under- or over-coverage is handled via the non-negative (slack) variables u_d and o_d, respectively.

The objective function

$$\min \sum_{e \in E, r \in R_e} c_{e,r}\, x_{e,r} + \sum_{d \in D} p_{u_d}\, u_d + \sum_{d \in D} p_{o_d}\, o_d$$

sums the costs $c_{e,r}$ for assigning roster r to employee e (which is used to model penalties for soft rule violations), and the penalties for under-coverage p_{u_d} and over-coverage p_{o_d} of all demands $d \in D$.

Solution Approach In the formulation above, the size of the sets of feasible rosters R_e is exponential in T. Thus, the number of roster variables $x_{e,r}$ is also exponential, and it is not possible to solve the MIP for real-world problem instances with standard mixed-integer programming solvers.

We therefore use a branch-and-price framework, which combines the ideas of branch-and-bound and column generation. Branch-and-price frameworks are popular for solving large-scale optimization problems of this kind and have already been successfully applied in various application areas over the last years [3].

For our application, we can model the pricing problem as a resource-constrained shortest path problem (RCSPP) in a graph with one layer for each day of the planning period: there is a node for each allowed activity (for which there is a positive demand), and the arcs describe feasible activity sequences. Further constraints are modeled by as additional resources along the arcs. An s-t-path in this graph then corresponds to a feasible roster. This is illustrated in Fig. 1.

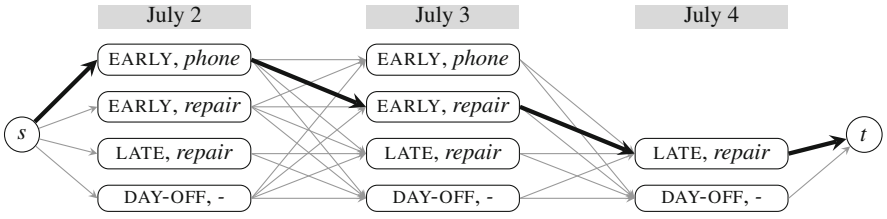


Fig. 1 Illustration of an exemplary RCSPP for "Peter" from the example of Sect. 1

For many customer scenarios, we observed that a large amount of the runtime is spent on solving the pricing problems. In particular for large instances with more than ten resources in the RCSPP, the exponential runtime of dynamic programming is a significant problem. Therefore, we have implemented several improvements.

Although an optimal solution to the RCSPP is needed as a stopping criterion in the column generation process, we do not necessarily need to find columns with minimum reduced costs. In fact, any column with negative reduced costs can be helpful in the current iteration of the column generation. Therefore, we initially apply a depth-first-search heuristic, which is able to quickly generate new columns in many cases. Furthermore, we implemented a cyclic column generation approach: in each iteration of the column generation, the pricing problems are only solved for a subset of the employees before the restricted master problem is resolved. In order to obtain integrality, we implemented constraint branching strategies.

The algorithm has been implemented in C++ and is based on the branch-cut-price framework BCP of the Computational Infrastructure for Operations Research (COIN-OR) project [6]. The restricted master problems are solved with CLP.

Evaluation on Nurse Rostering Instances A challenging application for roster generation in practice is the rostering of nurses in hospitals. In order to evaluate the quality of our algorithm, we therefore compared it with the best-known solutions for publicly available nurse rostering instances. We used instances that were provided for the Second International Nurse Rostering Competition (INRC-II), which took place from October 2014 until June 2015 [2].

While the final ranking in the competition was based on the solution quality on a set of *hidden instances* not known to the participants in advance, a set of *late instances* with similar structure was fully accessible for the participants before submission. To have a fair comparison, we therefore focus on the late instances for our performance evaluation. While we use for our customer scenarios parallel computing techniques, we restricted ourselves here to stick to the rules of the competition, where only a single core was permitted and a runtime limit was given.

Each instance consists of a sequence of weekly scheduling problems for a number of nurses with various skills and different coverage requirements. The task is to find for each day an assignment of nurses to shifts, such that skills and minimum staffing requirements are met. There is a hard constraint for the allowed shift types on consecutive days, and there are soft constraints for over-coverage, minimum/maximum number of consecutive assignments and days off, preferences, complete weekends, total number of assignments and total number of working week-ends.

Table 1 shows the best objective values among all algorithms submitted by the finalists of the competition and the solution of our algorithm on the late instances of the INRC-II. The results show that we could provide for 10 out of 28 instances better solutions than the ones reported by the finalists of the competition. We are able to generate competitive results on the remaining instances.

Table 1 Comparison of the objective values of our algorithm and the best solution among all algorithms submitted by the finalists of INRC-II (better solutions are highlighted)

Instance id	Best cost	Our cost	Ratio
n030w4_1_6-2-9-1	1755	1835	1.05
n030w4_1_6-7-5-3	1935	1950	1.01
n030w8_1_2-7-0-9-3-6-0-6	2340	**2330**	1.00
n030w8_1_6-7-5-3-5-6-2-9	1900	2000	1.05
n040w4_0_2-0-6-1	1730	1745	1.01
n040w4_2_6-1-0-6	1880	1920	1.02
n040w8_0_0-6-8-9-2-6-6-4	3310	3345	1.01
n040w8_2_5-0-4-8-7-1-7-2	2700	2980	1.10
n050w4_0_0-4-8-7	1490	1555	1.04
n050w4_0_7-2-7-2	1480	1535	1.04
n050w8_1_1-7-8-5-7-4-1-8	5410	5550	1.03
n050w8_1_9-7-5-3-8-8-3-1	5435	5545	1.02
n060w4_1_6-1-1-5	2815	**2730**	0.97
n060w4_1_9-6-3-8	2950	3065	1.04
n060w8_0_6-2-9-9-0-8-1-3	2765	**2760**	1.00
n060w8_2_1-0-3-4-0-3-9-1	3065	3170	1.03
n080w4_2_4-3-3-3	3535	3620	1.02
n080w4_2_6-0-4-8	3570	3570	1.00
n080w8_1_4-4-9-9-3-6-0-5	4995	**4805**	0.96
n080w8_2_0-4-0-9-1-9-6-2	5030	5280	1.05
n100w4_0_1-1-0-8	1445	**1410**	0.98
n100w4_2_0-6-4-6	2100	**1990**	0.95
n100w8_0_0-1-7-8-9-1-5-4	3080	**2840**	0.92
n100w8_1_2-4-7-9-3-9-2-8	3055	**2945**	0.96
n120w4_1_4-6-2-6	2435	**2320**	0.95
n120w4_1_5-6-9-8	2485	**2370**	0.95
n120w8_0_0-9-9-4-5-1-0-3	3615	3790	1.05
n120w8_1_7-2-6-4-5-2-0-2	3510	4165	1.19

3 Adapting and Integrating the Optimizer for Customers

There is a large variety among the processes and requirements for scheduling staff based on, for example, a country's laws, industry standards, and contractual agreements. For a successful application of our optimizer it is, thus, important that it can be adapted to a customer's needs. Interestingly, experience shows that in order to apply the optimizer in various scenarios, we don't need to modify the MIP formulation. The formulation is general enough to cover a large number of rostering requirements occurring in practice. The main challenge is to generate the input constraints and objectives for the optimizer (based on a customer's requirements and from the available data) as well as post-processing.

In order to do this in an easily-modifiable and modular fashion, we developed a domain-specific language (DSL) that allows us to both state business rules defining the mapping from a customer's data to the optimizer input, and access a customer's data in a database. Our DSL, called Roxx, is a dialect of Datalog, a declarative programming language that blends ideas from logic programming languages like Prolog with features from database query languages like SQL [1]. We start with an example on how to define the optimizer's input using Roxx's smallest building blocks, *deduction rules*, before giving an overview of the language and its implementation.

We consider an example where activities are defined in terms of shifts and functions. In order to specify the admissible rosters for an employee, we restrict the activities valid for an employee, such that (1) an activity is allowed if there is demand for it, (2) the employee can fulfill the requested function, and (3) the employee is not on vacation and, thus, can work the requested shift. In practice, we see that customer data is typically kept in tabular form. In our example, we have access to the following three tables:

DEMAND table

Date	Shift	Function
July 2	EARLY	Phone
July 2	EARLY	Repair
July 2	LATE	Repair
July 3	EARLY	Phone
July 3	EARLY	Repair
July 3	LATE	Repair
July 4	LATE	Repair

FUNCTION table

Employee	Function
Laura	Repair
Peter	Phone
Peter	Repair

VACATION table

Employee	Date
Laura	July 3

The tables provide data to determine the admissible rosters of the employees subject to the above conditions. In Roxx, we *model* the above condition by the following *deduction rule*:

```
ALLOWED-ACTIVITIES(employee, date, shift, function) ⊢
    DEMAND(date, shift, function),
    FUNCTION(employee, function),
    ¬ VACATION(employee, date).
```

Evaluating the rule means we consider the data provided for the input tables, match their entries according to the rule, and store the relevant attributes of the matching rows into the output table. For our example, we obtain the output table shown on next page, where Peter's rows describe the nodes of the RCSPP in Fig. 1.

Typically a Roxx model for a customer consists of hundreds of rules of the above kind where the output of one rule is the input to other rules. For example, instead of directly specifying functions, we may want to deduce them from employee qualifications using several rules. Moreover, Roxx extends standard Datalog by providing built-in language elements tailored to the needs of workforce

ALLOWED-ACTIVITIES table			
Employee	Date	Shift	Function
Laura	July 2	EARLY	*Repair*
Laura	July 2	LATE	*Repair*
Laura	July 3	DAY-OFF	–
Laura	July 4	LATE	*Repair*
Laura	July 4	DAY-OFF	–
Peter	July 2	EARLY	*Phone*
Peter	July 2	EARLY	*Repair*
Peter	July 2	LATE	*Repair*
Peter	July 2	DAY-OFF	–
...			
Peter	July 4	LATE	*Repair*
Peter	July 4	DAY-OFF	–

management. Evaluating a customer's model of deduction rules efficiently is done by a *deductive database* [7] that is implemented as part of our system.

4 Conclusion

We outlined our optimization algorithm for generating rosters, which is used to schedule staff in practice. We evaluated its performance on publicly-available nurse rostering data. Moreover, we described how the optimizer is integrated into our workforce management suite by using a domain-specific language that allows to easily describe the mapping from customer data to optimizer input.

References

1. Ceri, S., Gottlob, G., Tanca, L.: Logic Programming and Databases. Springer, New York (1990)
2. Ceschia, S., Dang, N., De Causmaecker, P., Haspeslagh, S., Schaerf, A.: The second international nurse rostering competition. Ann. Oper. Res. **274**, 171–186 (2018)
3. Desaulniers, G., Desrosiers, J., Solomon, M.M.: Column Generation. Springer, Berlin (2005)
4. Ernst, A.T., Jiang, H., Krishnamoorthy, M.: Staff scheduling and rostering: a review of applications, methods and models. Eur. J. Oper. Res. **153**, 3–27 (2004)
5. Kutschka, M., Herbers, J.: An insight to aviation: rostering ground personnel in practice. In Operations Research Proceedings 2014, pp. 349–355. Springer, Berlin (2016)
6. Lougee-Heimer, R.: The common optimization interface for operations research: promoting open-source software in the operations research community. IBM J. Res. Dev. **47**(1), 57–66 (2003)
7. Ramakrishnan, R., Gehrke, J.: Database Management Systems, 3rd edn. McGraw-Hill, New York (2003)

Scheduling in a Data Gathering Network to Minimize Maximum Lateness

Joanna Berlińska

1 Introduction

Scheduling for data gathering is becoming increasingly important due to a growing number of applications producing data in distributed locations. For example, data acquired by wireless sensor networks, or the results of computations executed in clouds or grids, are scattered on a number of processing nodes. These data usually need to be gathered on a single machine for merging and further processing. The efficiency of the data gathering process influences the performance of the whole application.

Scheduling in data gathering wireless sensor networks was studied in [6, 12]. The analyzed problem was to minimize the total sensing and data gathering time by assigning appropriate amounts of measured data to the network nodes. Later on, scheduling algorithms for gathering datasets of fixed sizes were proposed. Maximizing the network lifetime was analyzed in [1], and data gathering networks with data compression were studied in [2, 11]. Article [3] considered minimizing the maximum lateness in a network with defined dataset release times and due dates. In [1–3, 11], only transferring data to the base station was taken into account. Contrarily, [4] studied a network with limited base station memory, where datasets had not only to be gathered, but also processed by the base station.

In this work, we also analyze the two stages of data transfer and processing. Datasets have release times and due dates, and our goal is to minimize the maximum dataset lateness. The network works in a 2-machine flow shop mode. A review of flow shop scheduling problems with due date related criteria can be found in [13].

J. Berlińska (✉)
Faculty of Mathematics and Computer Science, Adam Mickiewicz University in Poznań, Poznań, Poland
e-mail: Joanna.Berlinska@amu.edu.pl

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_56

It should be noted that almost all results reported there, pertain to non-preemptive scheduling, whereas in our model, preemptions are allowed.

The rest of this paper is organized as follows. In Sect. 2 we formulate the scheduling problem. Its computational complexity is studied in Sect. 3. Algorithms for solving the problem are proposed in Sect. 4, and the results of computational experiments on their performance are presented in Sect. 5. The last section comprises conclusions.

2 Problem Formulation

The data gathering network consists of a set of m worker nodes $P_1, \ldots, P_m$ and a single base station. Node P_j produces dataset D_j of size α_j at time r_j. Each dataset has to be first sent directly to the base station, and then processed by it. The due date of dataset D_j is d_j. Transferring one unit of data to the base station takes time C, and processing this amount of data takes time A. At most one worker can communicate with the base station at a time. Both sending and processing a dataset can be preempted at any time and resumed later at no cost. Let T_j denote the time when processing of dataset D_j finishes. The lateness of dataset D_j is $L_j = T_j - d_j$. Our goal is to organize dataset transfers and processing so that the maximum dataset lateness $L_{max} = \max_{j=1}^{m}\{L_j\}$ is minimized. Thus, using the three-field notation, our problem can be denoted as $F2|r_j, pmtn, p_{1j} = C\alpha_j, p_{2j} = A\alpha_j|L_{max}$.

3 Complexity

It is known that problem $F2|r_j, pmtn|L_{max}$ is strongly NP-hard [5]. However, we solve its special case, where job processing times are not arbitrary. Therefore, we start this section with proving that our problem is also strongly NP-hard.

Proposition 1 *The analyzed scheduling problem is strongly NP-hard.*

Proof The proof is achieved by a pseudopolynomial reduction from the following strongly NP-complete 3-Partition problem [8]. Given a positive integer K and a set of $3n$ positive integers $\{x_1, \ldots, x_{3n}\}$ such that $K/4 < x_j < K/2$ for each $j \in \mathcal{N} = \{1, 2, \ldots, 3n\}$, and $\sum_{j=1}^{3n} x_j = nK$, is it possible to partition the index set $\mathcal{N}$ into n disjoint sets $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_n$ such that for each $1 \leq i \leq n$, $\sum_{j \in \mathcal{N}_i} x_j = K$?

Given an arbitrary instance of 3-Partition, we construct the following instance of our scheduling problem. Let $m = 4n + 1$, $C = A = 1$. There are $3n$ "regular" datasets with $\alpha_j = x_j$, $r_j = K$, $d_j = (2n + 1)K$, for $j = 1, \ldots, 3n$, and $n + 1$ "blocker" datasets with $\alpha_j = K$, $r_j = 2(j - 3n - 1)K$, $d_j = 2(j - 3n)K$, for $j = 3n + 1, \ldots, 4n + 1$. We will show that a schedule with $L_{max} \leq 0$ exists if and only if the corresponding instance of 3-Partition is a "yes"-instance.

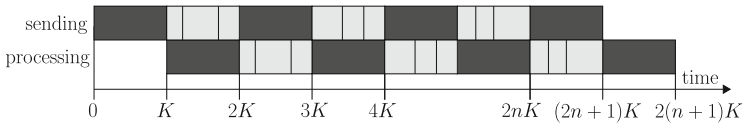


Fig. 1 Schedule structure for the proof of Proposition 1. Regular datasets are light gray, blocker datasets are dark gray

Let us first assume that the required 3-partition exists. We construct the following schedule (see Fig. 1). Blocker datasets D_j (where $j > 3n$) are transferred in intervals $[r_j, r_j + K)$ and processed in intervals $[r_j + K, d_j)$. Regular datasets D_j such that $j \in \mathcal{N}_i$, are transferred one by one in interval $[(2i-1)K, 2iK)$, and processed in interval $[2iK, (2i+1)K)$, for $i = 1, \ldots, n$. The maximum lateness in this schedule is $L_{max} = 0$.

Conversely, suppose that the required 3-partition does not exist. In any schedule with $L_{max} \leq 0$ each blocker dataset D_j has to be transferred in interval $[r_j, r_j + K)$, and processed in interval $[r_j + K, d_j)$. Moreover, there can be no computation idle time in interval $[K, 2(n+1)K]$. Let $i \leq n$ be the smallest positive number such that the total size of regular datasets whose transfers complete in interval $[(2i-1)K, 2iK]$ is smaller than K. Such a number exists, because otherwise we would have a "yes" answer to the analyzed instance of 3-Partition problem. The total size of datasets fully transferred before time $(2i+1)K$ (when the next blocker dataset arrives at the base station) is smaller than $2iK$. Hence, the total size of data that can be processed in interval $[K, (2i+1)K)$ is smaller than $2iK$, which means that we have an idle time in processing in this interval. This precludes constructing a schedule with $L_{max} \leq 0$.

Thus, we showed that a schedule with $L_{max} \leq 0$ exists if and only if the corresponding 3-partition exists, which completes the proof. □

Although our scheduling problem is computationally hard in general, the following special cases can be solved in polynomial time. If $C = 0$ or $A = 0$, our problem is equivalent to problem $1|pmtn, r_j|L_{max}$, and can be solved in $O(m \log m)$ time by the preemptive EDD rule [9]. If $r_j = r$ and $d_j = d$ for $j = 1, \ldots, m$, then our problem reduces to a special case of problem $F2|pmtn|C_{max}$, and can be solved in $O(m \log m)$ time by Johnson's algorithm [10]. Finally, if $C \leq A$ and $d_j = d$ for $j = 1, \ldots, m$, then our problem becomes a special case of problem $F2|1\text{-}min, pmtn, r_j|C_{max}$, and can be solved in $O(m \log m)$ time by the preemptive shortest remaining transfer time rule [7].

4 Algorithms

Let us note that the difficulty in solving our problem lies in constructing the communication schedule. For a given communication schedule, computing the optimum processing schedule consists in solving problem $1|r_j, pmtn|L_{max}$, which

can be done in $O(m \log m)$ time. Thus, all the proposed algorithms concentrate on constructing a communication schedule.

First, we describe an exact branch-and-bound algorithm (BB). BB is a standard technique for solving combinatorial optimization problems. A branching rule divides the set of possible solutions until distinguishing unique solutions. Bounding rules eliminate sets of solutions which are infeasible or certainly not better than some already known solution. In our problem one has to construct a sequence of communications defined by triplets of numbers: start time, dataset number, and transfer time. A leading sequence of communications is extended by appending at its end a transfer of an available (i.e. already released, but not yet fully transferred) dataset. The transfer length is chosen as the minimum of two values: the time necessary to complete the dataset, and the time until the nearest dataset release. If there is no available dataset, the communication network remains idle until the next dataset release. Branching is limited by the following two rules. Firstly, the transfer of dataset D_j can preempt sending some other dataset only at time r_j. Secondly, if the transfer of dataset D_i is preempted by D_j, then sending D_i cannot be resumed until the transfer of D_j is completed. It can be proved using the interchange argument that an optimum schedule abiding to these rules can always be found. In order to bound the enumeration of possible solutions, for each partial communication sequence, a partial computation schedule is constructed, taking into account only the datasets that already arrived at the base station. If the maximum lateness obtained for this schedule is not smaller than the best L_{max} value found so far, the partial solution is discarded. The initial upper bound on L_{max} is the smallest value found by the heuristics described below.

We analyze three greedy heuristics for constructing the communication sequence. Algorithm FIFO transfers datasets in the order in which they were released. Thus, no communication preemptions occur. Heuristic EDD chooses an available dataset with the smallest due date, and preempts an earlier dataset transfer, if necessary. Finally, algorithm SRT selects an available dataset with the shortest remaining transfer time, possibly preempting an earlier communication. Each of these heuristics runs in $O(m \log m)$ time.

5 Computational Experiments

The performance of our algorithms was tested in computational experiments. Due to limited space, we report here only on a small subset of obtained results. In the test instances presented here, the number of datasets was $m = 10$, due to the high complexity of the BB algorithm. The network parameters were $A, C \in \{1, 2, 3, 4, 5\}$. The release time of the first dataset was $r_1 = 0$. The remaining release times were computed from the formula $r_j = r_{j-1} + \delta_j$, where δ_j was chosen randomly from interval [1, 10], for each j independently. Dataset due dates were selected randomly from interval [0, 100], and dataset sizes α_j from interval [1, 15]. As a measure of heuristic solutions quality we used the absolute error,

Table 1 Average solution errors vs. C, for $A = 1$

Algorithm	$C = 1$	$C = 2$	$C = 3$	$C = 4$	$C = 5$
FIFO	18.82	47.50	46.25	40.55	45.34
EDD	0.78	1.13	0.75	1.72	0.99
SRT	18.95	43.84	45.39	42.85	40.48

Table 2 Average solution errors vs. A, for $C = 1$

Algorithm	$A = 1$	$A = 2$	$A = 3$	$A = 4$	$A = 5$
FIFO	18.82	4.13	1.13	0.97	0.96
EDD	0.78	2.81	4.21	3.29	2.06
SRT	18.95	6.01	1.44	1.61	1.09

i.e. the difference between the obtained maximum lateness and the optimum value computed by BB. Using a relative measure was not possible, as the optimum value of L_{max} could be zero or negative. For each analyzed setting, 100 instances were generated and solved.

Table 1 presents the average errors obtained for $A = 1$ and changing $C \geq A$. It can be seen that algorithm EDD delivers very good schedules, whereas FIFO and SRT perform similarly to each other, and produce much worse results. Changing C from 1 to 2 causes a significant increase in the errors of FIFO and SRT schedules, but increasing C to more than 2 does not have a strong impact.

The results for $C = 1$ and different values of $A \geq C$ are presented in Table 2. Increasing A makes algorithms FIFO and SRT work better, and for $A \geq 3$ they both outperform EDD. The errors of EDD solutions first increase, and then decrease. Indeed, when A is very big, the communication part of the schedule becomes less important in comparison to the computation part, which we solve to the optimum. Thus, all algorithms produce good results in this case.

6 Conclusions

In this paper, we analyzed minimizing the maximum dataset lateness in a data gathering network. The scheduling problem was proved to be strongly NP-hard. We proposed an exact branch-and-bound algorithm and three polynomial-time greedy heuristics. The results of computational experiments show that heuristic EDD obtains very good solutions. However, when processing is 3 or more times slower than communication, EDD is outperformed by FIFO and SRT algorithms.

Future research may include improving the branch-and-bound algorithm by adding new pruning rules. Another interesting direction is analyzing the case when resuming dataset transfer after preemption introduces additional time cost.

Acknowledgements This research was partially supported by the National Science Centre, Poland, grant 2016/23/D/ST6/00410.

References

1. Berlińska, J.: Communication scheduling in data gathering networks with limited memory. Appl. Math. Comput. **235**, 530–537 (2014). https://doi.org/10.1016/j.amc.2014.03.024
2. Berlińska, J.: Scheduling for data gathering networks with data compression. Eur. J. Oper. Res. **246**, 744–749 (2015). https://doi.org/10.1016/j.ejor.2015.05.026
3. Berlińska, J.: Scheduling data gathering with maximum lateness objective. In: Wyrzykowski, R., et al. (eds.) Parallel Processing and Applied Mathematics: 12th International Conference PPAM 2017, Part II. LNCS, vol. 10778, pp. 135–144. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-78054-2_13
4. Berlińska, J.: Scheduling data gathering with limited base station memory. In: Caramia, M., et al. (eds.) Proceedings of the 16th International Conference on Project Management and Scheduling, pp. 42–45. TexMat, Rome (2018)
5. Cho, Y., Sahni, S.: Preemptive scheduling of independent jobs with release and due times on open, flow and job shops. Oper. Res. **29**, 511–522 (1981). https://doi.org/10.1287/opre.29.3.511
6. Choi, K., Robertazzi, T.G.: Divisible load scheduling in wireless sensor networks with information utility. In: IEEE International Performance Computing and Communications Conference 2008: IPCCC 2008, pp. 9–17 (2008)
7. Cheng, J., Steiner, G., Stephenson, P.: A computational study with a new algorithm for the three-machine permutation flow-shop problem with release times. Eur. J. Oper. Res. **130**, 559–575 (2001). https://doi.org/10.1016/S0377-2217(99)00415-4
8. Garey, M.R., Johnson, D.S.: Computers and Intractability, a Guide to the Theory of NP-Completeness. W. H. Freeman, Hayden (1979)
9. Horn, W.A.: Some simple scheduling algorithms. Naval Research Logistics Quarterly **21**, 177–185 (1974). https://doi.org/10.1002/nav.3800210113
10. Johnson, S.M.: Optimal two- and three-stage production schedules with setup times included. Naval Research Logistics Quarterly **1**, 61–68 (1954). https://doi.org/10.1002/nav.3800010110
11. Luo, W., Xu, Y., Gu, B., Tong, W., Goebel, R., Lin, G.: Algorithms for Communication Scheduling in Data Gathering Network with Data Compression. Algorithmica **80** (2017). https://doi.org/10.1007/s00453-017-0373-6
12. Moges, M., Robertazzi, T.G.: Wireless sensor networks: scheduling for measurement and data reporting. IEEE Trans. Aerosp. Electron. Syst. **42**, 327–340 (2006)
13. Pérez González, P., Framiñán Torres, J.M., González, P.L., León Blanco, J.M., Ruiz Usano, R.: Flowshop scheduling problems with due date related objectives: A review of the literature. In: 3rd International Conference on Industrial Engineering and Industrial Management, pp. 1488–1497 (2009)

Heuristics for Solving the Job Sequencing and Tool Switching Problem with Non-identical Parallel Machines

Dorothea Calmels, Chandrasekharan Rajendran, and Hans Ziegler

1 Introduction

Modern manufacturing systems have to provide different module functions such as milling, drilling and turning, or tool and work-piece handling operations to efficiently deal with product variety and changing customer requirements. Each machine is equipped with a tool magazine of limited capacity that should hold at least the number of tools needed for processing a single job. Sequencing of product variations is a challenging task, since different operations require different sets of tools. In general, the number of tools required to process all jobs exceeds the tool magazine capacity, such that tools must be brought to and from the tool storage to the local machine magazine, and tool switches may be required within the machine magazine between two consecutive jobs. The setup time of a machine for a job is the time span needed for tool switching. If the tools cannot be interchanged during job processing then switching time becomes especially crucial when the switching time is significant in regard to the processing time of a job. The challenge of the job sequencing and tool switching problem (SSP) is to find the best job sequence such that, e.g., the number of tool switches is minimized. The uniform SSP for a single machine has first been addressed by [1] who showed that the connected sub-problem is solvable in polynomial-time by the 'Keep Tool Needed Soonest' (KTNS) policy. The SSP was proven to be NP-hard by [2].

D. Calmels (✉) · H. Ziegler
University of Passau, Passau, Germany
e-mail: dorothea.calmels@uni-passau.de; hans.ziegler@uni-passau.de

C. Rajendran
Indian Institute of Technology Madras, Chennai, India
e-mail: craj@iitm.ac.in

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_57

This paper is directed to the SSP for multiple non-identical parallel machines (SSP-NPM). The SSP-NPM is defined by three interdependent problems: (1) assign a set of jobs to each of the machines, and (2) sequence the jobs on the machines, while in the process, (3) defining the tool loading so that a given objective is optimized. Non-identical implies that all of the machines can execute the same operations but may have different magazine capacities and different tools, and the processing time of a job need not be equal for all machines.

The SSP for multiple machines was first mentioned by [3] for tandem machines. Heuristics for the SSP for identical parallel machines have been addressed by [4] who conclude that the problem is NP-hard. [5] present a mixed-integer nonlinear program for machines with identical working conditions but different magazine capacities. [6] extend the IP formulation of [1, 7] for identical parallel machines and given processing times with the objective of minimizing makespan. [8, 9] investigate the SSP for unrelated parallel machines without considering the magazine capacity but including a limited amount of tool copies to minimize the makespan. So far, the SSP-NPM with time aspects has not been addressed in literature. However, related problems for sequencing PCBs exist. [10, 11] consider the scheduling of printed circuit packs on multiple parallel sequencers that are equipped with a number of dispensing heads that have to be loaded with the required input tapes.

The remainder of this article is as follows. The problem definition is given in Sect. 2, followed by the description of the heuristics for the SSP-NPM in Sect. 3. Computational results are presented in Sect. 4. Section 5 concludes the study.

2 Problem Statement

For the SSP-NPM, a given set of jobs J has to be processed on a set of non-identical parallel machines M. Each machine m is theoretically able to process all jobs and sufficient tools are available to process all jobs on all machines. Only one job j can be processed on each machine at a time k and each job has to be assigned to exactly one machine. The processing of the jobs on each machine happens sequentially. The term non-identical means here that the processing times p_{jm} of any job j can depend on the machine m, and that the machines may have different magazine capacities C_m. K is a set of time periods k, needed to include timetabling in the extended model. A time instant k is measured at the end of period k. T denotes the set of tools that is required for processing all jobs. It is assumed that each tool occupies only one slot in the tool magazine of a machine. Each job requires a subset of tools T_j that has to be present in the tool magazine before this job is processed and J_t denotes the subset of jobs that require a certain tool t. However, the tool magazine capacity C_m of any machine m cannot hold the tools necessary for processing all jobs, so that tool

switches may become necessary for two successive jobs. A tool switch is defined as the removal of a tool from its slot and inserting another tool in the same place [1]. It is assumed that tools cannot be changed during the processing. The time to switch any tool on a machine m is defined as sw_m.

The SSP-NPM consists of simultaneously finding the assignment of jobs to machines as well as the right tool loading. The objective in general is to find for each machine the set of jobs to be processed and their sequence, such that the total number of tool switches (TS) is minimized. In reality, not only the number of tool switches but different time-related objectives have to be considered like minimizing total flowtime (TFT) or minimizing makespan ($Fmax$). These objectives additionally require the completion time f_j of job j for all jobs. Table 1 presents the parameters of an illustrative example with 6 jobs, 9 tools and 2 machines with $C_1 = 4$ and $C_2 = 3$, and $sw_1 = 1$ and $sw_2 = 2$. The solution for minimizing TS is illustrated in Fig. 1.

Table 1 Example: tools used for each job and processing times p_{j1} and p_{j2} (in time periods), magazine capacities $C_1 = 4$ and $C_2 = 3$

Jobs	1	2	3	4	5	6
Tools	1	1	2	1	3	1
	4	3	6	5	5	2
	8	5	7	7	8	4
	9		8	9		
p_{j1}	1	3	4	5	6	1
p_{j2}	2	4	3	5	3	1

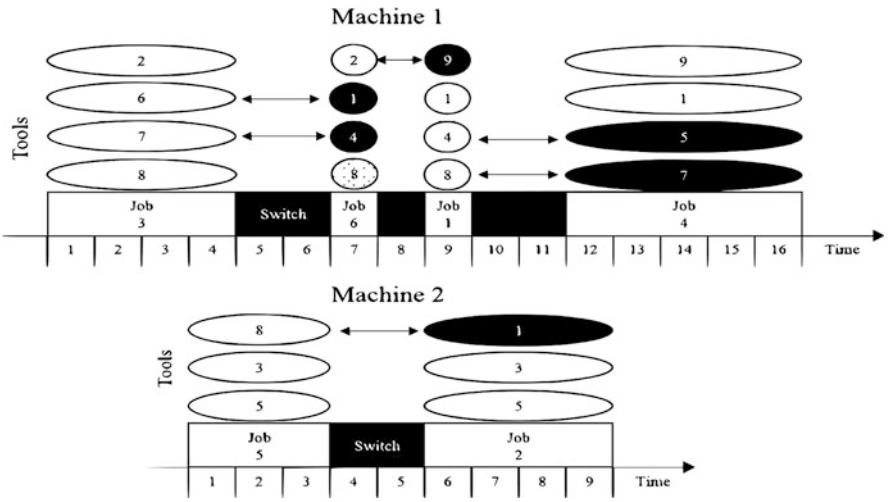


Fig. 1 Solution to the initial example with arrows indicating tool switches; min $TS = 6$

3 Heuristics for the SSP-NPM

3.1 Structure of the Heuristics

Each heuristic generates a job sequence for each machine. Therefore, let index [j] denote the number of the job found in position j in the sequence Π_m of jobs processed on machine m. Let i_m denote the approximated completion time of the last job on machine m. Ties are always broken so that the element with the lowest index is selected.

	Algorithm 1: basic heuristic scheme
1	initialize $\Pi_m = \emptyset$, $i_m = 0$
2	**for** each machine **do**
3	*generate.sequence*(i_m, r) Π_m according to *assignment_rule* $r \in \rho$
4	*generate.tool_load*(J_t, T_j, Π_m) with KTNS-policy
5	**end for**

3.2 Rules for Assignment

In this section, a set of three simple assignment rules ρ is presented, simple to the extent that tool switching is not considered during the process. Each *assignment_rule* $r \in \rho$ is used to generate different job sequences for each machine. The basic scheme for generating the job sequence for any rule is presented below (*Algorithm 2*). The selection of the jobs itself is depending on the assignment rule.

	Algorithm 2: basic scheme for *generate.sequence*(i_m, r)
1	initialize *J_left* = J,
2	**while** *J_left* $\neq \emptyset$ **do**
3	*m.selection*(i_m) // select 'free' machine, *m_selected* $\leftarrow$ argmin$_m(i)$
4	*j.selection(r, m_selected)* // select the job, *j_selected* $\in$ *J_left* according to rule r,
5	$\Pi_m \leftarrow \Pi_m \cup \{j\_selected\}$ // append *j_selected* to the sequence on the free machine
6	$i_m \leftarrow i_m + p_{j\_selected,m\_selected}$ // update i
7	*J_left* $\leftarrow$ *J_left* $\setminus \{j\_selected\}$ // remove selected job
8	**end while**

SPT-Based Heuristic ($r = 1$) The shortest processing time (SPT)-based heuristic requires jobs being sorted in non-descending order of their processing times for all machines while satisfying the capacity restrictions. On the machine which is available earliest, the job with the smallest processing time is selected for assignment: $j\_selected \leftarrow argmin_j\ (p_{j,m\_selected})$.

LPT-Based Heuristic ($r = 2$) The longest processing time (LPT)-based heuristic requires jobs being sorted in non-ascending order of their processing times for all machines while satisfying the capacity restrictions. On the machine which is available earliest the job with the highest processing time is selected for assignment: $j\_selected \leftarrow argmax_j\ (p_{j,m\_selected})$.

GI-Heuristic ($r = 3$) The greatest intersection (GI) requires the number of common tools between two possible consecutive jobs in a job sequence on a machine. At the beginning of the sequences, the jobs with the highest number of tools is selected. Subsequently, on the selected machine the job with the greatest intersection of tools compared to the previously processed job is selected for assignment on that machine.

	Algorithm 3: basic scheme for *j.selection (r = 3, m_selected)*
1	**if** $i_m = 0$ **then**
2	$j\_selected \leftarrow \text{argmax}_j\ \lvert T_j \rvert$
3	**end if**
4	**else if** $i_m > 0$ **then**
5	$intersect_j = (\lvert T_j \cap T_{[j\text{-}1]} \rvert)$
6	$j\_selected \leftarrow \text{argmax}_j(intersect_j)$
7	**end if**

4 Computational Results

Experiments were conducted on a 2.50 GHz Intel Core i7-4710 processor with 16 GB of memory running under Windows 8. RStudio 1.1.453 with R 3.5.1 was used to implement and solve the heuristics. The instances of [6] were used but had been modified for multiple non-identical parallel machines by adding machine specific processing and tool switching times. Each problem group consists of 120 instances for the small ($|J| \leq 25$) instances and 60 instances for the large ($|J| \geq 50$) instances. A total of 1440 small instances and 1440 large instances had been tested.

Table 2 summarizes the results and shows the percentage of finding the best known solution per heuristic. Overall, the mean CPU time was 0.05 s for the SPT and LPT heuristics and 0.47 s for the GI-heuristic. Even large instances could be solved below 1 s.

The results confirm that the SPT-based heuristic performs well for minimizing *TFT* or *Fmax*. The LPT-based heuristic performs worst for all objectives and the GI-heuristic performs best for minimizing *TS* and well for minimizing time-related objectives especially for large instances. Yet, the SPT-based heuristic outperforms the GI-based heuristic in terms of solution quality for small problem instances.

Table 2 Average percentage (%) of finding the best known solution for *TS*, *TFT* and *Fmax*

	SPT-based heuristic			LPT-based heuristic			GI-heuristic		
Problem type	TS	TFT	Fmax	TS	TFT	Fmax	TS	TFT	Fmax
Small	10.8	**84.7**	**58.8**	9.6	0.6	3.3	**88.5**	14.8	38.1
Large	3.5	48.8	44.4	0.2	0.1	0.0	**96.3**	**51.1**	**55.6**

Note: Boldface numbers indicate the maximum percentage among the heuristics

5 Conclusion

This paper presents simple and fast heuristics for the SSP-NPM with timetabling. Small and large instances were solved for different objective functions. The steadiest performance is achieved by the GI-heuristic for minimizing the number of tool switches while the SPT heuristic works rather well for minimizing total flow time or makespan. Further investigations and statistics tests are required in order to investigate the influence of the problem instance specifications on the solution quality. The quality of the solutions seems to differ significantly for variations in the relation between processing time and tool switching time but also variation of the job-tool matrix density. Future research may integrate the tool setup time as reducing the tool switches becomes especially important when the switching time is significant in regard to the processing time of a job. Overall, the heuristics seem to provide promising initial solutions for further local search procedures or metaheuristics.

References

1. Tang, C.S., Denardo, E.V.: Models arising from a flexible manufacturing machine, part i: minimization of the number of tool switches. Oper. Res. **36**(5), 767–777 (1988)
2. Crama, Y., Kolen, A.W.J., Oerlemans, A.G., Spieksma, F.C.R.: Minimizing the number of tool switches on a flexible machine. Int. J. Flex. Manuf. Syst. **6**(1), 33–54 (1994)
3. Bard, J.F.: A heuristic for minimizing the number of tool switches on a flexible machine. IIE Trans. **20**(4), 382–391 (1988)
4. Fathi, Y., Barnette, K.W.: Heuristic procedures for the parallel machine problem with tool switches. Int. J. Prod. Res. **40**(1), 151–164 (2002)
5. Sarmadi, H., Gholami, S.: Modeling of tool switching problem in a flexible manufacturing cell: with two or more machines. In: Lee, G. (ed.) International Conference on Mechanical and Electrical Technology, 3rd, (ICMET-London 2011), vol. 1–3, pp. 2345–2349. ASME, New York (2011)
6. Beezão, A.C., Cordeau, J.-F., Laporte, G., Yanasse, H.H.: Scheduling identical parallel machines with tooling constraints. Eur. J. Oper. Res. **257**(3), 834–844 (2017)
7. Laporte, G., Salazar-González, J.J., Semet, F.: Exact algorithms for the job sequencing and tool switching problem. IIE Trans. **36**(1), 37–45 (2004)
8. Özpeynirci, S., Gökgür, B., Hnich, B.: Parallel machine scheduling with tool loading. Appl. Math. Model. **40**(9–10), 5660–5671 (2016)

9. Gökgür, B., Hnich, B., Özpeynirci, S.: Parallel machine scheduling with tool loading: a constraint programming approach. Int. J. Prod. Res. **56**(16), 5541–5557 (2018)
10. Ghrayeb, O.A., Phojanamongkolkij, N., Finch, P.R.: A mathematical model and heuristic procedure to schedule printed circuit packs on sequencers. Int. J. Prod. Res. **41**(16), 3849–3860 (2003)
11. Van Hop, N., Nagarur, N.N.: The scheduling problem of PCBs for multiple non-identical parallel machines. Eur. J. Oper. Res. **158**(3), 577–594 (2004)

Solution Algorithm for Time/Cost Trade-off Stochastic Project Scheduling Problem

Takumi Kitamura and Takayuki Shiina

1 Introduction

In project scheduling, the project manager considers deadline, budget, and available resources. There are two project scheduling models, which involve resources. One is the Resource Constrained Project Scheduling Problem (RCPSP). The RCPSP has restrictions on the amount of resources that can be used concurrently. The other is the Time/Cost Trade-off Problem (TCTP).

Zhu et al. [6] presented a two-stage stochastic programming model for project planning. We propose a new stochastic programming model for TCTP and solution to the problem.

In this paper, the jobs involved in the project are assumed to be handled by people. The amount of job is defined using man-hour. The total amount of workload required to complete a job is called a man-hour. By increasing the number of people inputted to the job, the duration of the job can be shortened. The cost increases according to the number of the people; therefore, in order to reduce the cost, we must reduce the number of people. However, the duration of the project will be longer. In other words, cost and time are determined by the amount of resources devoted to job. The number of resources is assumed to be discrete. This model is referred to as the Discrete Time/Cost Trade-off Problem (DTCTP). One of the typical methods is the Critical Path Method (CPM). Conventional research on DTCTP can be classified into the following categories.

T. Kitamura · T. Shiina (✉)
Waseda University, Okubo, Shinjuku-ku, Tokyo, Japan
e-mail: t-kitamura@asagi.waseda.jp; tshiina@waseda.jp

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_58

Deadline problem (DTCTP-D)
Given the deadline of the project, minimize the total cost as an objective function (Öncü et al. [4]).

Budget problem (DTCTP-B)
Minimize the make-span to the extent that the given budget is not exceeded (Öncü [3] and Değirmenci et al. [2]).

Time/Cost curve type problem (DTCTP-C)
A model expressing the relationship between budget and deadline as a curve. This study belongs to this model. Studies that take into account the experience curve effect of a job were presented by Jeang [5].

There is limited research on project scheduling considering man-hour. In this study, man-hour is defined as the product of number of people and the duration as (man-hour) = (number of people) × (duration). We have to consider various types of uncertainties in the execution of the project. In the case of manual work by people, it is difficult for the project manager to estimate quantitatively the progress of the job. There is a possibility that delays and budget overrun may occur with respect to the determined schedule. In this study, this problem is formulated by stochastic programming. The first stage variable is defined as the amount of resources devoted. After the realization of random duration time is observed, the second stage decision must be made to minimize the penalty cost. The penalty for exceeding the deadline can be calculated by reducing the time with respect to the fluctuation of the job time, which is called crashing.

2 Formulation of the Problem

In the formulation of the problem, we will use the following notation.

Parameters

P	Precedence set
N	Set of jobs (The last is dummy job representing completion of the project)
S	Set of scenarios
M_i	Man-hour of job i
L_i	Lower bound for resources devoted to job i
U_i	Upper bound for resources devoted to job i
π^s	Probability of scenario s
q_i^s	Unit cost of additional resource input to job i under scenario s
p	Unit penalty cost when makespan exceeds the deadline
a_i	Unit crashing time when additional resources are inputted to job i
D	Deadline for project
A_i	Parameter concerning the lower bound of the duration of job i
B_i	Parameter concerning the upper bound of the crashing time of job i ($A_i \geq B_i$)

Random Variables

ξ_i^s Ratio of fluctuation for the duration of job i under scenario s
d_i^s The duration of job i under scenario $s(d_i^s = \lceil d_i \xi_i^s \rceil, d_i^s \in Z_+)$
d_i Standard duration of job i $(d_i \in Z_+^{|N|})$

The decision variables are defined as follows.
First-Stage Decision Variables

x_i The amount of resources inputted to job i

Second-Stage Decision Variables

t_i^s Time needed to start job i under scenario s
y_i^s Number of additional resources inputted to job i under scenario s

In this research, we aim to minimize the total cost. The objective function can be shown as follows.

$$\min \boldsymbol{M}\boldsymbol{x} + Q(\boldsymbol{x}) \tag{1}$$

$$Q(\boldsymbol{x}) = \sum_{s=1}^{S} \pi_s v_s(x, y) \tag{2}$$

$$L_i \le x_i \le U_i (i = 1, \ldots, |N|) \tag{3}$$

$$x_{|N|} = 0, \boldsymbol{x} \in Z_+^{|N|} \tag{4}$$

The first term in (1) represents the total cost concerning the number of people inputted. This term is calculated by multiplying the unit cost c of a job and its duration d. The value of c increases by $O(x^2)$ for x number of people [1]. For simplicity, we set $c = x^2$ for x number of people inputted. Therefore, the cost concerning personnel expenses can be expressed as $cd = x^2 \frac{M}{x} = Mx$.

In the second stage problem, penalty for crashing and exceeding the deadline are minimized for the decision variable y.

$$v_s(x, y) = \min \; p(t_{|N|}^s - D)^+ + \sum_{i=1}^{|N|} q_i y_i^s \tag{5}$$

$$\text{subject to } \; t_j^s - t_i^s \ge d_i^s - a_i y_i^s \ge \frac{M_i}{A_i} \; (i, j) \in P \tag{6}$$

$$d_i^s \ge d_i \xi_i^s \ge \frac{M_i}{x_i} \xi_i^s \; (i = 1, \ldots, |N|) \tag{7}$$

$$0 \le y_i^s \le \frac{M_i}{B_i} \; (i = 1, \ldots, |N|) \tag{8}$$

$$y_{|N|} = 0, \; \boldsymbol{y}, \; \boldsymbol{d^s} \in Z_+^{|N|} \tag{9}$$

Constraint (7) is a nonlinear inequality. In order to transform the constraint (7) to the linear inequality, the following M_i valid inequalities are used for each job.

$$d_i^s \geq -\frac{M_i \xi_i^s}{k(k+1)} x_i + \frac{2M_i \xi_i^s k + M_i \xi_i^s}{k(k+1)}, k = 1, 2, \cdots, M_i \tag{10}$$

Inequalities (6) and (7) can be expressed as follows.

$$t_j^s - t_i^s \geq -\frac{M_i \xi_i^s}{k(k+1)} x_i + \frac{2M_i \xi_i^s k + M_i \xi_i^s}{k(k+1)} - a_i y_i^s, k = 1, 2, \cdots, M_i \tag{11}$$

3 Numerical Examples

For the numerical experiments, the following network has been selected. The priority relationship of jobs in a project is represented in Fig. 1 as Job-On-Node (J-O-N). The number of nodes represents the number of the jobs. The man-hour M_i for job i is displayed above the node. Node 7 is a dummy job indicating the completion of the project. We assume that the probability fluctuations occur independently in Job 3 and Job 4. Furthermore, there are two scenarios of realization for the random duration. In order to introduce uncertainty, we consider the case in which independent job 3 and 4 each has 50% chance of being 3 times as long as its standard duration ($\xi_3^1 = \xi_4^1 = 1$. $\xi_3^2 = \xi_4^2 = 3$). Therefore, there are 4 scenarios. We assume that the random variables follow *beta* distribution. The typical probability density function is shown as follows.

$$f(x) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)}, \quad B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} dt \tag{12}$$

By using the numerical integration, the value of probability is computed. We set $\alpha = 2, \beta = 5$. In this case, the probability takes maximum value at 0.2. So we regard 0.2 as the most likely value. We set half value 0.1 and three times value 0.6 as the optimistic value and the pessimistic value.

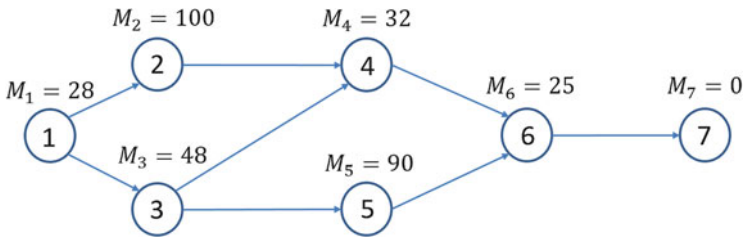


Fig. 1 J-O-N Network

Table 1 Results of numerical experiments at $p = 200$

Deadline	RP			EEV			VSS[%]
	First stage	Recourse	Total cost	First stage	Recourse	Total cost	
15	2767	2012.5	4779.5	2967	2250	5217	8.38605
30	2201	525	2762	2291	950	3241	15.8902
45	1861	0	1861	1655	1050	2705	31.2015
60	1437	0	1437	1231	1400	2631	45.382

Under these conditions, multiple patterns of deadline are given. The schedules of the stochastic programming model and the deterministic equivalent model are compared.

The unit cost q for additional shortened time y (crashing) is set to 150, which is larger than the coefficient M_i of the first stage decision variable. The value of a_i is set to 2. This means that the job is shortened by 2 h per additional resource input. The lower bound L_i and the upper bound U_i are set to 1 and 10, respectively, for all x_i. The parameters concerning the additional input of resources are given as $A_i = 25$ and $B_i = 20$ for all i. The penalty values for exceeding the deadline is set to 100 and 200.

Table 1 presents the results of the two models, the stochastic programming model (RP) and the deterministic equivalent model (EEV). The longer the deadline D, the smaller the number of people working x. Figure 2 illustrates two graphs for the results of the numerical experiments for the cases of $p = 100$ and $p = 200$. The deadline value was given in five increments from 15 to 60. Figure 2 shows the relationship of trade-off between the total cost and the deadline.

Figure 3 shows a Gantt chart of scheduling when the deadline is 35. The horizontal axis represents time and the vertical axis represents the amount of devoted resources. In this SDTCTP model, the additional input for each scenario is permitted and the expected deadline penalty and additional decision cost are incorporated into the objective function. As a result, the schedule varies for each scenario.

In scenario 2, one unit of additional resource is added to jobs 2 and 4. In scenario 3, one additional resource is devoted to job 3. In scenario 4, one additional resource is added to jobs 1 and 4. The DTCTP model is a schedule created using the determination problem with the expected value. Because the minimum number of resources so as not to exceed the deadline are introduced, there is a risk that a penalty will occur.

4 Concluding Remarks

In this research, we have shown the stochastic programming model considering the man-hour, which is an important factor in project scheduling. An effective solution, using linear model for DTCTP-C with probability variation, was developed. In the

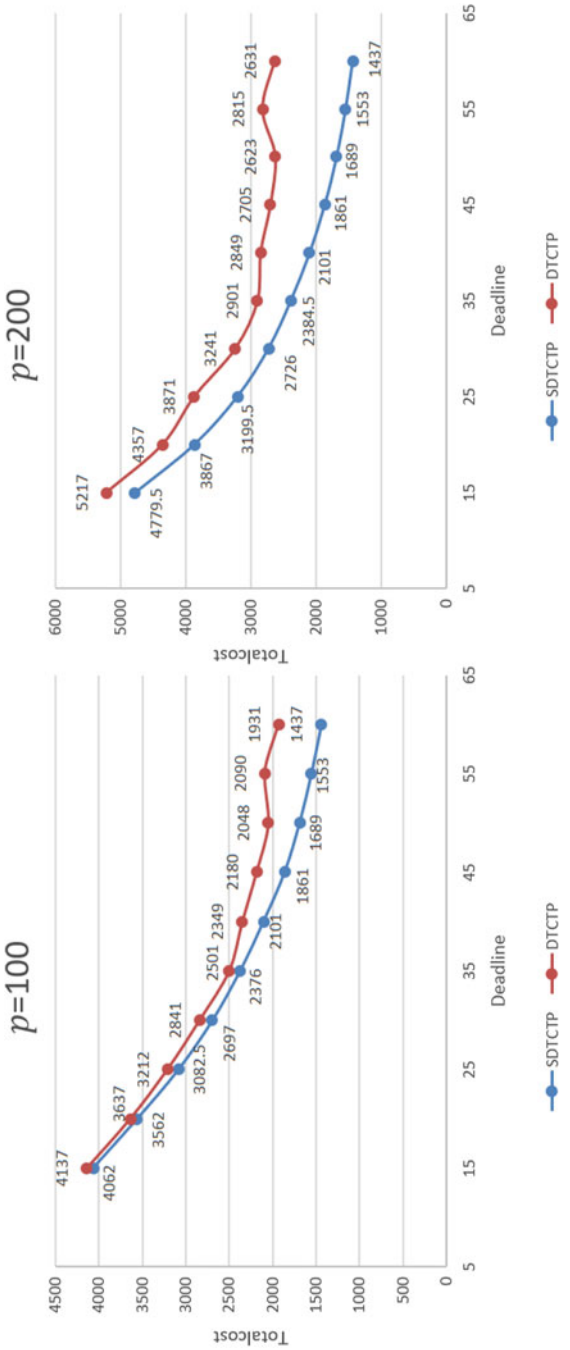


Fig. 2 Time/Cost Trade-off Curve

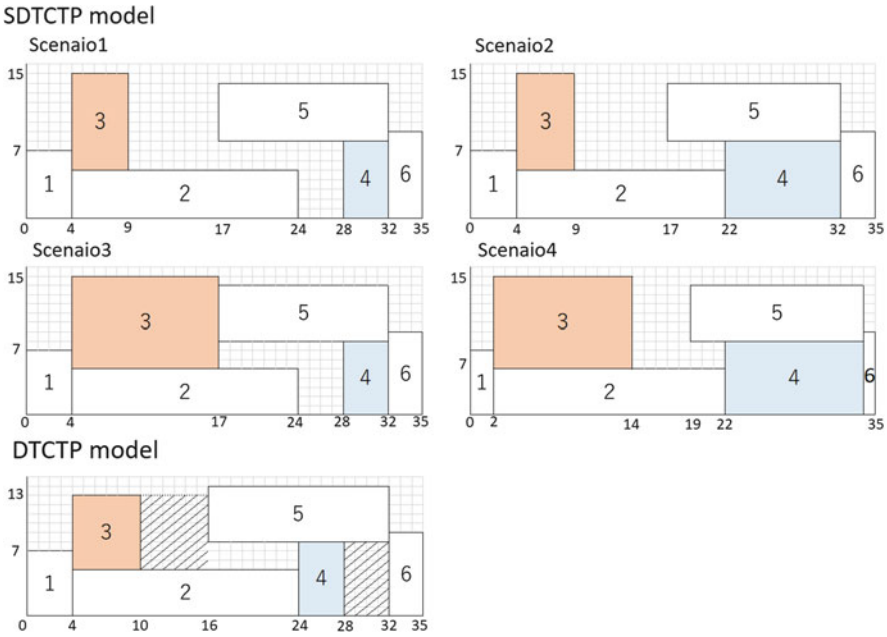


Fig. 3 Gantt chart (Deadline 35)

case of a project with a large number of jobs, the number of scenarios becomes enormous. In the future, we will consider adopting a scenario reduction method using moment matching.

References

1. Brooks Jr, F.P.: The Mythical Man-Month: Essays on Software Engineering, Anniversary Edition, 2nd edn. Addison-Wesley Professional, Boston (1995)
2. Değirmenci, G., Azizoğlu, M.: Branch and bound based solution algorithms for the budget constrained discrete time/cost trade-off problem. J. Oper. Res. Soc. **64**, 1474–1484 (2013)
3. Hazir, Ö., Haouari, M., Erel, E.: Discrete time/cost trade-off problem: A decomposition-based solution algorithm for the budget version. Comput. Oper. Res. **37**, 649–655 (2010)
4. Hazir, Ö., Erel, E., Günalay, Y.: Robust optimization models for the discrete time/cost trade-off problem. Int. J. Prod. Econ **130**, 87–95 (2011)
5. Jeang, A.: Project management for uncertainty with multiple objectives optimisation of time,cost and reliability. Int. J. Prod. Res. **53**(5), 1503–1526 (2015)
6. Zhu, G., Bard, J.F., Yu, G.: A two-stage stochastic programming approach for project planning with uncertain activity durations. J. Sched. **10**, 167–180 (2007)

Complexity and Approximation Results for Setup-Minimal Batch Scheduling with Deadlines on a Single Processor

Dominik Kress, Maksim Barketau, Erwin Pesch, and David Müller

1 Introduction

Consider the problem of having to sequence n jobs on a single processor (machine). Each job belongs to one of F nonempty *families*. A family solely includes jobs that are sufficiently similar to allow their processing without the need for an intermediate setup operation. Family F_f, $f \in \{1, \ldots, F\}$, consists of n_f jobs, so that $n = \sum_{f=1}^{F} n_f$. We denote the j-th job of family F_f, $f \in \{1, \ldots, F\}$, by (j, f). Each job (j, f) has a *processing time* $p_{j,f} \in \mathbb{N}$ and a *deadline* $d_{j,f} \in \mathbb{N}$. Jobs may not be preempted. We assume that the jobs of each family are sorted and indexed in non-decreasing order of their respective deadlines. Jobs with identical deadlines within a family are arranged in the order of increasing indices. When consecutively processing jobs of a single family, these jobs are said to form a *batch* of the schedule. At the beginning of the schedule and between the processing of two batches, a *setup* operation is required. These setup operations are associated to a setup cost $s \in \mathbb{N}$ but are assumed to not require time. The objective is to schedule all jobs such that they are completed no later than by their respective deadlines and such that the total setup cost is minimized, which is identical to minimizing the number of setup operations.

D. Kress (✉) · D. Müller
Management Information Science, University of Siegen, Siegen, Germany
e-mail: dominik.kress@uni-siegen.de; david.mueller@uni-siegen.de

M. Barketau
United Institute of Informatics Problems, NAS of Belarus, Minsk, Belarus
e-mail: barketau@mail.ru

E. Pesch
Management Information Science, University of Siegen, Siegen, Germany

Center for Advanced Studies in Management, HHL Leipzig, Leipzig, Germany
e-mail: erwin.pesch@uni-siegen.de; erwin.pesch@hhl.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_59

We will refer to this problem as the *batch scheduling problem with setup costs and deadlines* (BSPCD) for the remainder of this paper.

Overviews of scheduling problems that involve setup considerations are given in [1–4, 10]. BSPCD was first considered by Bruno and Downey [5]. The authors proved the decision version of BSPCD to be NP-complete, while it remained open whether it is strongly NP-complete. In this paper, we answer this question by providing a sketch of a proof for strong NP-completeness of the decision version of BSPCD. The full version of this proof is presented in [8]. We additionally provide an $O(n \log n + nF)$ algorithm that approximates the cost of an optimal schedule by a factor of F. The proof of this approximation ratio is given in [8].

2 Computational Complexity

We now present a pseudo-polynomial transformation [7] from the strongly NP-complete 3-Partition problem to the decision version of BSPCD, which asks if there exists a feasible schedule with a total scheduling cost of no more than a given K. Our proof is based on results presented in [6] and [9]. Its full version is presented in [8]. The main ideas are summarized in this section.

3-Partition is defined as follows [7]: Given $3m + 1$ integers $u_1, \ldots, u_{3m}$, B with $\sum_{i=1}^{3m} u_i = mB$ and $\frac{B}{4} < u_i < \frac{B}{2}$ $\forall i = 1, \ldots, 3m$. Does there exist a partition of the set $\{1, \ldots, 3m\}$ into m subsets $U_1, \ldots, U_m$, such that $\sum_{i \in U_j} u_i = B$ $\forall j = 1, \ldots, m$?

The following claim can be proven by making use of some simple job shifting and interchange arguments. It will play a central role in the proof of Theorem 1.

Lemma 1 *Let I be an instance of BSPCD. If there exists a feasible schedule for I, then there exists an optimal schedule S with the jobs of each family being processed in nondecreasing order of their deadlines (EDD order).*

Theorem 1 *The decision version of BSPCD is strongly NP-complete.*

Proof The fact that the decision version of BSPCD is in NP is easy to see.

Given an instance I_P of 3-Partition, we construct an instance I_B of the decision version of BSPCD in polynomial time as follows. We set $K = (7m + 1)s$, with $s \in \mathbb{N}$ arbitrary. The number of jobs n is set to $3m(m + 1) + 2m + 1$. We consider $F = 4m + 1$ families. We furthermore define distinct deadlines

$$D_j = (2j - 1)(X + Y) + \frac{3m(m+1)}{2} Z + \frac{3(j-1)(2m-j+2)}{2} Z + (m + 1) \sum_{i=1}^{j-1} B + \sum_{i=j}^{m} iB$$

for $1 \leq j \leq m + 1$. For $f = 1, \ldots, 3m$, we set $F_f = \{(j, f) | 1 \leq j \leq m + 1\}$ with $p_{j,f} = u_f + Z$ and $d_{j,f} = D_j$ for all $(j, f) \in F_f$. Similarly, for $f = 3m + 1, \ldots, 4m$, we set $F_f = \{(j, f) | 1 \leq j \leq 2\}$ with $p_{j,f} = X + Y$ and $d_{j,f} = D_{f-3m-1+j}$ for all $(j, f) \in F_f$. Finally, for $f = 4m + 1$, we set $F_f = \{(1, f)\}$,

$p_{1,f} = X + Y$, and $d_{1,f} = D_{m+1}$. Here, X, Y, and Z are arbitrary, nonnegative integers with $X > \frac{1}{2}(m^2-m)B$, $Z > \frac{1}{4}(m^2+m)B$, and $Y \geq \frac{3}{2}(m^2-m)Z$. Observe that family F_f relates to u_f, $f \in \{1, \ldots, 3m\}$.

In the following, we will sketch a potential way of showing that I_P is a yes-instance if and only if I_B is a yes-instance. We will refer to all jobs with deadline D_j, $j \in \{1, \ldots, m+1\}$, as D_j-jobs. Furthermore, given a schedule for I_B, we will denote the completion time of the last D_j-job within this schedule by C_j, $j \in \{1, \ldots, m+1\}$. We will restrict ourselves to the nontrivial case $m > 1$.

Assume that I_P is a yes-instance. Then it is possible to re-label the indices of $u_1, \ldots, u_{3m}$ and their corresponding families in I_B, such that $u_{3i-2}+u_{3i-1}+u_{3i} = B$ for $1 \leq i \leq m$. Now, let each family F_f, $f \in \{3m+1, \ldots, 4m+1\}$, act as a batch and divide all remaining families F_f, $f \in \{1, \ldots, 3m\}$, into exactly two batches $B_f = \left\{(j, f) | 1 \leq j \leq \left\lceil \frac{f}{3} \right\rceil \right\}$ and $A_f = F_f \setminus B_f$. Construct a schedule S by sequencing these batches as follows:

$$S = B_1, B_2, \ldots, B_{3m}, F_{3m+1}, A_1, A_2, A_3, F_{3m+2}, \ldots, A_{3m-2}, A_{3m-1}, A_{3m}, F_{4m+1}.$$

The jobs within each batch are sequenced in their EDD order. It is easy to see that this requires $7m+1$ setup operations so that the total scheduling cost is K. Furthermore, one can show that $C_j = D_j$ for $1 \leq j \leq m+1$. Hence, all jobs are on time and we have established a solution to I_B.

Now assume that I_B is a yes-instance and let S be a feasible schedule with a total scheduling cost of no more than K that satisfies the EDD property of Lemma 1. Then, for each batch of a family F_f, $f \in \{1, \ldots, 4m+1\}$, in S and any two jobs (i, f) and (j, f) with $i < j$ that are included in the batch, all jobs of the set $\{(k, f) | i < k < j\}$ are included in the batch as well. We denote the unique batch of family F_f, $f \in \{1, \ldots, 3m\}$, that includes a D_1-job by B_f and re-label the indices of $F_1, \ldots, F_{3m}$ and their corresponding integer numbers of I_P, such that $|B_1| \leq |B_2| \leq \cdots \leq |B_{3m}|$. Hence, $B_f = \{(1, f), \ldots, (|B_f|, f)\}$ for $f \in \{1, \ldots, 3m\}$.

In [8], we provide proofs for four properties. Let S be constructed as described above. Then:

- $C_j < C_{j+1}$ for all $j \in \{1, \ldots, m\}$.
- For all $j \in \{1, \ldots, m\}$, S can be transformed into a feasible schedule S', where the jobs $(1, 3m+j)$ and $(2, 3m+j)$ are processed without intermediate setup and where the number of setup operations does not increase. We assume that this transformation has been performed on S for the remainder of this proof.
- For all $j \in \{1, \ldots, m\}$, $(1, 3m+j)$ is the last D_j-job that is processed before C_j.
- For all $j \in \{1, \ldots, m\}$, there must exist exactly $l_j := \sum_{i=j}^{m} 3(m+1-i)$ jobs that belong to families F_f, $f \in \{1, \ldots, 3m\}$, and that have not been started to be processed at time C_j in S.

Based on these properties, we may assume that each family F_f, $f = 3m + 1, \ldots, 4m$, is processed in one batch in S, while each family F_f, $f = 1, \ldots, 3m$, is divided into exactly two batches, so that $|B_{3i-2}| = |B_{3i-1}| = |B_{3i}|$ for all $i \in \{1, \ldots, m\}$, $B_f = \left\{(j, f) | 1 \leq j \leq \left\lceil \frac{f}{3} \right\rceil \right\}$, and $A_f := F_f \setminus B_f$ defines a batch of S. Now, denote the sum $u_{3i-2} + u_{3i-1} + u_{3i}$ by μ_i for all $i \in \{1, \ldots, m\}$. Then, by definition of 3-Partition, $\sum_{i=1}^{m} \mu_i = mB$. Furthermore, based on the above deliberations, we have $C_j = (2j - 1)(X + Y) + \sum_{i=1}^{m} i(3Z + \mu_i) + \sum_{i=1}^{j-1}(m - i + 1)(3Z + \mu_i)$ for all $j \in \{1, \ldots, m\}$. By additionally demanding $C_j \leq D_j$ for all $j \in \{1, \ldots, m\}$ and by applying a Lemma given in [9], we can show that this results in $\mu_1 = \cdots = \mu_m = B$, so that we have established a solution to I_P. □

3 Approximation Algorithm

We will now restrict our attention to instances with $d_{j,f} - d_{j-1,f} \geq p_{j,f}$ for all $f \in \{1, \ldots, F\}$, $j \in \{2, \ldots, n_f\}$. As shown in [8], any given instance of BSPCD can be modified into an equivalent instance that fulfills this property in polynomial time. We will further consider some specific classes of schedules:

- A general *EDD-schedule* is a job sequence in which the jobs are processed in non-decreasing order of their deadlines. Ties among jobs are broken in favor of the job with the smaller family index or, if this index is identical, the smaller job index. Hence, there is a unique EDD-schedule for a given problem instance.
- In a group technology schedule (*GT-schedule*), one requires precisely F setups in the schedule, so that each family must form exactly one batch.
- A *GTD-schedule* is a specific GT-schedule, in which the families F_f, $f = 1, \ldots, F$, are processed in non-decreasing order of the values $d_{1,f} + T_f - p_{1,f}$, where $T_f := \sum_{j=1}^{n_f} p_{j,f}$. Ties among families are broken by favoring the family with the smaller family index. The jobs within each batch are processed in their EDD order with ties being broken as described above.

We observe:

Lemma 2 *Let I be an instance of BSPCD. There exists a feasible schedule for I if and only if the EDD-schedule of I is feasible.*

However, the EDD-schedule may, in general, provide a fairly bad approximation of the optimal schedule.

Lemma 3 *An EDD-schedule for an instance of BSPCD can have $n/2$ times more batches than a corresponding optimal schedule.*

With respect to GT- and GTD-schedules, one can show:

Lemma 4 *Let I be an instance of BSPCD. There exists a GT-schedule that is feasible (and thus optimal) for I if and only if the GTD-schedule is feasible for I.*

The proofs of the above lemmas are given in [8].

For the sake of notational convenience in the remainder of this paper, define $n_0 := 0$ and denote job (j, f), $f \in \{1, \ldots, F\}$, $j \in \{1, \ldots, n_f\}$, with the index $n_0 + \cdots + n_{f-1} + j$. Assume the jobs to be ordered in non-decreasing order of their deadlines with ties being broken in favor of smaller job indices. Now, consider an instance I of BSPCD. For given $i, k \in \{1, \ldots, n\}$, $i \leq k$, we can define a related instance I', that solely considers jobs $i, \ldots, k$ of I. The processing times of the jobs remain unchanged, while the deadlines are set to their value in I minus the total processing time of jobs 1 to $i - 1$. We refer to problem BSPCD on I' as the (i, k)-*problem* of I. The proof of the following lemma is given in [8].

Lemma 5 *Let I be an instance of BSPCD and consider any (i, k)-problem of I. If the GTD-schedule for the (i, k)-problem is feasible, then the GTD-schedule is feasible for every (l, k)-problem, where $i < l \leq k$.*

The following algorithm, that returns a schedule σ^0, is based on these results:

0. Initialization: Set $k := n$, $\sigma := \emptyset$, and $\sigma^0 := \emptyset$.

1. GTD-schedule: Determine the GTD-schedule σ^{GTD} of the input instance. If σ^{GTD} is feasible, then it is also optimal. In this case, set $\sigma^0 = \sigma^{GTD}$ and stop.

2. EDD-schedule: Determine the EDD-schedule σ^{EDD} of the input instance. If σ^{EDD} is infeasible, then there exists no feasible schedule. In this case, stop.

3. Stop criterion: If $k \leq F$, set $\sigma^{max} := (1, \ldots, k)$, $\sigma^0 := (\sigma^{max}, \sigma)$ and stop. Else, set $i := k - F$ and $\sigma^{max} := (i + 1, \ldots, k)$.

4. (i, k)-problem: Determine the GTD-schedule σ^{GTD} for the (i, k)-problem.

Case 1, σ^{GTD} is feasible: Set $\sigma^{max} := \sigma^{GTD}$. If $i = 1$, set $\sigma^0 := (\sigma^{max}, \sigma)$ and stop. Else, if $i > 1$, set $i := i - 1$ and repeat Step 4.

Case 2, σ^{GTD} is infeasible: Set $\sigma := (\sigma^{max}, \sigma)$, $k := i$, and go to Step 3.

Step 1 of the algorithm applies Lemma 4, while steps 2 and 4 apply Lemmas 2 and 5, respectively. Our main result regarding this algorithm is proven in [8].

Theorem 2 *Let I be an instance of BSPCD. The above algorithm checks if there exists a feasible schedule for I. If a feasible schedule exists, it determines a schedule that approximates the cost of an optimal schedule by a factor of F.*

In a computational study in our full paper [8], we show that the approximation algorithm, on average, performs significantly better than indicated by its approximation ratio. In order to provide more insights in this paper, we have implemented a greedy heuristic as an additional benchmark for the approximation algorithm. It iteratively schedules the jobs by first selecting the family of the last job of the current schedule (or, in the first iteration, the set of all jobs) and afterwards selecting one of the family's (or set's) jobs based on a given priority rule (EDD or shortest processing time, SPT). This job is scheduled next if the overall schedule that results from tentatively scheduling all remaining jobs in their EDD order after this job remains feasible. If this is not the case, the algorithm schedules a job with smallest deadline out of the remaining jobs. This guarantees the construction of feasible schedules.

Table 1 Computational results—average and maximum quality ratios

F	n	Approx. alg.	Greedy (EDD)	Greedy (SPT)
10	45	2.67 (3.4)	2.50 (3)	2.87 (3.4)
20	140	3.31 (3.7)	3.41 (3.8)	4.31 (4.7)
30	285	3.86 (4.3)	4.24 (4.8)	5.66 (6.27)
40	480	4.24 (4.63)	4.96 (5.63)	7.08 (7.50)
50	725	4.66 (5.34)	5.42 (5.94)	8.43 (8.68)

Our testbed is composed of 5 groups of instances with $F \in \{10, 20, 30, 40, 50\}$. Each group features 20 randomly generated instances with integer processing times and deadlines. For details, we refer to [8]. The computational results (all algorithms implemented in C++) are presented in Table 1. It depicts the average values and maximum values (in parentheses) of the quality ratios returned by each heuristic. The quality ratio of a solution corresponds to the objective function value of this solution divided by the number of families F (lower bound on the optimal objective function value). The results indicate that the approximation algorithm, on average, results in better solutions than the greedy algorithm for both priority rules and $F \geq 20$. With respect to computational times, all algorithms terminate within a few milliseconds.

References

1. Allahverdi, A.: The third comprehensive survey on scheduling problems with setup times/costs. Eur. J. Oper. Res. **246**(2), 345–378 (2015)
2. Allahverdi, A., Gupta, J.N.D., Aldowaisan, T.: A review of scheduling research involving setup considerations. Omega **27**(2), 219–239 (1999)
3. Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov M.Y.: A survey of scheduling problems with setup times or costs. Eur. J. Oper. Res. **187**(3), 985–1032 (2008)
4. Błażewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Węglarz, J.: Handbook on Scheduling: From Theory to Applications. Springer, Berlin (2007)
5. Bruno, J., Downey, P.: Complexity of task sequencing with deadlines, set-up times and changeover costs. SIAM J. Comput. **7**(4), 393–404 (1978)
6. Cheng, T.C.E., Ng, C.T., Yuan, J.J.: The single machine batching problem with family setup times to minimize maximum lateness is strongly NP-hard. J. Sched. **6**(5), 483–490 (2003)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability – A Guide to the Theory of NP-Completeness. Freeman, New York (1979)
8. Kress, D., Barketau, M., Pesch, E.: Single-machine batch scheduling to minimize the total setup cost in the presence of deadlines. J. Sched. **21**(6), 595–606 (2018)
9. Lu, L.F., Yuan, J.J.: The single machine batching problem with identical family setup times to minimize maximum lateness is strongly NP-hard. Eur. J. Oper. Res. **177**(2), 1302–1309 (2007)
10. Potts, C.N., Kovalyov, M.Y.: Scheduling with batching: a review. Eur. J. Oper. Res. **120**(2), 228–249 (2000)

Exact and Heuristic Solution Approaches for a Flexible Job Shop Scheduling Problem Incorporating Machine Operator Restrictions

David Müller, Dominik Kress, and Jenny Nossack

1 Introduction

A well known class of scheduling problems that represents many real-world manufacturing systems is the *job shop scheduling problem* (JSP), see e.g. [5]. A generalization of the JSP, referred to as the *flexible job shop scheduling problem* (FJSP), takes into account the fact that manufacturing systems oftentimes feature multiple machines of the same type as well as multi-purpose machines that allow for processing different types of operations. In the FJSP, each operation must be processed by exactly one machine out of a given set of *eligible machines* [6, 9]. Moreover, in real-world manufacturing systems, *setup times* play an important role [1–4] and the workforce planning must be integrated into the decision making process, especially when having to consider differently skilled *machine operators* (workers) [7]. In this paper, we therefore address a scheduling problem that combines these aspects. We refer to this problem as the FJSP with sequence-dependent setup times and differing worker skills and denote it by WSFJSP. WSFJSP is strongly NP-hard. Details are presented in our full paper [10].

D. Müller (✉) · D. Kress
Management Information Science, University of Siegen, Siegen, Germany
e-mail: david.mueller@uni-siegen.de; dominik.kress@uni-siegen.de

J. Nossack
Center for Advanced Studies in Management, HHL Leipzig, Leipzig, Germany
e-mail: jenny.nossack@googlemail.com

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_60

2 Notation and Model Formulation

The WSFJSP is defined as follows. Given is a set $I = \{1, \ldots, n\}$ of jobs and a set M of machines. Each job $i \in I$ is associated with an ordered set of q_i non-preemptive operations $O_i = (i_1, \ldots, i_{q_i})$. For any pair of operations $i_j, i_k \in O_i$, $i \in I$, with $j < k$, i_j must be completed before the processing of i_k may start. Each operation $i_j \in O_i$, $i \in I$, must be processed by exactly one machine out of a set of eligible machines $M_{i_j} \subseteq M$. To simplify the notation, we define $M_{i_j,k_l} := M_{i_j} \cap M_{k_l}$ for all $i, k \in I$, $i_j \in O_i$, $k_l \in O_k$. A worker out of a given set W must be assigned to an operation during its entire processing. Each machine and each worker can process at most one operation at a time. The completion time of an operation $i_j \in O_i$ of job $i \in I$ is denoted by C_{i_j}. The completion time of job $i \in I$ is denoted by C_i. A job is completed if all of its operations are completed. Hence, $C_i = C_{i_{q_i}}$ for all $i \in I$. The processing time $p_{i_j}^{m,w} \in \mathbb{Q}^+$ of an operation $i_j \in O_i$ of a job $i \in I$ is assumed to vary over different machines $m \in M_{i_j}$ and workers $w \in W$. We set $p_{i_j}^{m,w} = \infty$ if worker $w \in W$ is not eligible for processing an operation $i_j \in O_i$ of a job $i \in I$ on machine $m \in M_{i_j}$. Additionally, we assume that for each operation $i_j \in O_i$ of a job $i \in I$ and each corresponding machine $m \in M_{i_j}$, there exists at least one worker $w \in W$ that can process the operation within finite time. Sequence-dependent setup times $s_{i_j,k_l}^m \in \mathbb{Q}_0^+$ occur when an operation $k_l \in O_k$, $k \in I$, is processed immediately after an operation $i_j \in O_i$, $i \in I$, on machine $m \in M_{i_j,k_l}$. We assume that setup-operators are not scarce, so that setup operations do not require the assignment of a worker.

The problem is to assign each operation to one of its eligible machines, to sequence the operations on the machines, and to determine a corresponding assignment of workers to operations, such that the makespan $C_{max} := \max_{i \in I} C_i$ is minimized.

In order to allow the modelling of the first operation on each machine, we define a dummy job 0 with exactly one operation 0_1 with $M_{0_1} = M$. We set $M_{0_1,i_j} = M_{i_j,0_1} = M_{i_j}$ for all $i \in I$ and $i_j \in O_i$. Moreover, we set $p_{0_1}^{m,w} = 0$ for all $m \in M$ and $w \in W$. The setup times s_{0_1,i_j}^m can take arbitrary nonnegative rational values for all $m \in M$, $i \in I$, and $i_j \in O_i$. Furthermore, $s_{i_j,0_1}^m = 0$ for all $m \in M$, $i \in I$, and $i_j \in O_i$.

As motivated in [10], the WSFJSP can be addressed from a vehicle routing perspective. We therefore make use of a (vertex) set $V := \bigcup_{i \in I} O_i \cup \{0_1\}$ and define $\bar{V} := V \setminus \{0_1\}$, $V_{i_j} := V \setminus \{i_k | k \leq j\}$, $\tilde{V}_{i_j} := V \setminus \{i_k | k \geq j\}$, and $\bar{V}_{i_j} := V_{i_j} \setminus \{0_1\}$ for all $i_j \in V$. Now, for all $i_j \in V$, we define a continuous variable $t_{i_j} \in \mathbb{R}_0^+$ that represents the time when operation i_j is started to be processed on one of the

machines, and a continuous variable $C_{max} \in \mathbb{R}_0^+$ that represents the makespan. Moreover, we define the following binary variables:

$$y_{i_j,k_l}^m := \begin{cases} 1, & \text{if } k_l \text{ is processed directly after} \\ & i_j \text{ on } m \\ 0, & \text{else} \end{cases} \qquad \forall\, i_j \in V, k_l \in V_{i_j}, m \in M_{i_j,k_l}, \tag{1}$$

$$x_{i_j}^{m,w} := \begin{cases} 1, & \text{if } i_j \text{ is processed by } w \text{ on } m \\ 0, & \text{else} \end{cases} \qquad \forall\, i_j \in \bar{V}, m \in M_{i_j}, w \in W, \tag{2}$$

$$x_{i_j,k_l}^w := \begin{cases} 1, & \text{if } i_j \text{ and } k_l \text{ are processed by } w \\ 0, & \text{else} \end{cases} \qquad \forall\, i_j, k_l \in \bar{V}, i_j \neq k_l, w \in W, \tag{3}$$

$$v_{i_j,k_l} := \begin{cases} 1, & \text{if the processing of } k_l \text{ starts before the} \\ & \text{processing of } i_j \text{ finishes} \\ 0, & \text{else} \end{cases} \qquad \forall\, i_j, k_l \in \bar{V}, i_j \neq k_l. \tag{4}$$

Let B be a large positive number and define $p_{i_j}^{m,min} := \min_{w \in W} p_{i_j}^{m,w}$ for all $i \in I \cup \{0\}$, $i_j \in O_i$, and $m \in M_{i_j}$. Furthermore, set $p_{i_j}^{min} := \min_{m \in M_{i_j}} p_{i_j}^{m,min}$ for all $i \in I \cup \{0\}$ and $i_j \in O_i$. Then, a mixed-integer program (MIP) for WSFJSP is as follows.

$$\min C_{max} \tag{5}$$

s.t.

$$t_{i_{q_i}} + \sum_{m \in M_{i_{q_i}}} \sum_{w \in W} x_{i_{q_i}}^{m,w} \cdot p_{i_{q_i}}^{m,w} \leq C_{max} \qquad \forall\, i \in I, \tag{6}$$

$$\sum_{i_j \in \bar{V}_{k_l}} \sum_{m \in M_{i_j,k_l}} y_{i_j,k_l}^m = 1 \qquad \forall\, k_l \in \bar{V}, \tag{7}$$

$$\sum_{i_j \in \bar{V}} y_{0_1,i_j}^m \leq 1 \qquad \forall\, m \in M, \tag{8}$$

$$\sum_{k_l \in \bar{V}_{i_j} : m \in M_{k_l}} y_{k_l,i_j}^m - \sum_{k_l \in V_{i_j} : m \in M_{k_l}} y_{i_j,k_l}^m = 0 \qquad \forall\, i_j \in V, m \in M_{i_j}, \tag{9}$$

$$t_{i_j} + s^m_{i_j,k_l} + \sum_{w \in W} x^{m,w}_{i_j} \cdot p^{m,w}_{i_j} - t_{k_l} \leq \left(1 - y^m_{i_j,k_l}\right) B \quad \forall\, i_j \in V, k_l \in \bar{V}_{i_j}, m \in M_{i_j,k_l}, \tag{10}$$

$$t_{i_j} + \sum_{m \in M_{i_j}} \sum_{w \in W} x^{m,w}_{i_j} \cdot p^{m,w}_{i_j} \leq t_{i_{j+1}} \quad \forall\, i_j \in \bar{V} \text{ with } j \leq q_i - 1, \tag{11}$$

$$t_{i_j} + p^{min}_{i_j} \leq t_{i_{j+1}} \quad \forall\, i_j \in \bar{V} \text{ with } j \leq q_i - 1, \tag{12}$$

$$x^w_{i_j,k_l} \geq \sum_{m \in M_{i_j}} x^{m,w}_{i_j} + \sum_{m \in M_{k_l}} x^{m,w}_{k_l} - 1 \quad \forall\, i_j, k_l \in \bar{V}, i_j \neq k_l, w \in W, \tag{13}$$

$$t_{i_j} + \sum_{m \in M_{i_j}} \sum_{w \in W} x^{m,w}_{i_j} \cdot p^{m,w}_{i_j} - t_{k_l} \leq B \cdot v_{i_j,k_l} \quad \forall\, i_j, k_l \in \bar{V}, i_j \neq k_l, \tag{14}$$

$$x^w_{i_j,k_l} \leq 2 - v_{i_j,k_l} - v_{k_l,i_j} \quad \forall\, i_j, k_l \in \bar{V}, i_j \neq k_l, w \in W, \tag{15}$$

$$\sum_{i_j \in \bar{V}_{k_l} : m \in M_{i_j}} y^m_{i_j,k_l} = \sum_{w \in W} x^{m,w}_{k_l} \quad \forall\, k_l \in \bar{V}, m \in M_{k_l}, \tag{16}$$

$$y^m_{i_j,k_l} \in \{0,1\} \quad \forall\, i_j \in V, k_l \in V_{i_j}, m \in M_{i_j,k_l}, \tag{17}$$

$$x^{m,w}_{i_j} \in \{0,1\} \quad \forall\, i_j \in \bar{V}, m \in M_{i_j}, w \in W, \tag{18}$$

$$x^{m,w}_{0_1} = 0 \quad \forall\, m \in M_{i_j}, w \in W, \tag{19}$$

$$x^w_{i_j,k_l} \in \{0,1\} \quad \forall\, i_j, k_l \in \bar{V}, i_j \neq k_l, w \in W, \tag{20}$$

$$v_{i_j,k_l} \in \{0,1\} \quad \forall\, i_j, k_l \in \bar{V}, i_j \neq k_l, \tag{21}$$

$$t_{i_j} \in \mathbb{R}^+_0 \quad \forall\, i_j \in V, \tag{22}$$

$$C_{max} \in \mathbb{R}^+_0. \tag{23}$$

The objective function (5) minimizes the makespan of the schedule. Constraints (6) set a lower bound on the makespan. Constraints (7) ensure that each operation is scheduled exactly once, while inequalities (8) guarantee that there is at most one operation that is processed first on each machine. Flow conservation is enforced by constraints (9). A time increase of at least $s^m_{i_j,k_l} + p^{m,w}_{i_j}$ (setup and processing) compared to t_{i_j} if $y^m_{i_j,k_l} = 1$ is enforced by constraints (10). Constraints (11) guarantee that an operation $i_{j+1} \in O_i$ of job $i \in I$ can only start when its preceding operation $i_j \in O_i$ has been processed completely. Constraints (12) are redundant to constraints (11), but have shown to improve the computational performance in our tests. Inequalities (13) and (14) ensure that the variables (3) and (4) are set to one when needed. Based on these variables, constraints (15)

guarantee that each worker is assigned to at most one operation at a time. The connection of the sequencing variables with the worker assignment variables is enforced by constraints (16). Finally, constraints (17)–(23) define the domains of the variables.

3 Decomposition Based Solution Approaches

In [10], we propose to solve the WSFJSP in a branch-and-cut framework by decomposing it into a vehicle routing problem (VRP) with precedence constraints (master problem, MP), which—roughly speaking—focusses on a variation of constraints (7)–(12), and a worker assignment problem (subproblem), mainly taking constraints (13)–(16) into account. The MP explicitly addresses the allocation of operations to machines and the sequencing of operations on each machine. The effect of the assignment of workers to operations on the objective function value is embedded into the MP by making use of logic inequalities. Within our branch-and-cut framework, these inequalities are consecutively obtained by the subproblem, which explicitly determines an assignment of workers to operations such that the makespan is minimized based on a given allocation and sequencing decision of (a relaxed version of) the MP.

We furthermore present a heuristic approach that is based on this decomposition. In the first step, which follows an algorithmic idea of Giffler and Thompson [8], this approach iteratively allocates (and sequences) operations to the machines. In each iteration, it selects a machine and considers all operations that can start being processed at the respective point of time when considering all precedence constraints. Among all operations that compete for the same machine in some iteration, exactly one operation is chosen based on a priority rule. Given the resulting initial allocation and sequencing decisions, the second step determines a feasible worker assignment using a *beam search* approach. This approach constructs a search tree in a breadth-first search manner by assigning workers to the operations which are ordered in non-decreasing order of the points in time when their processing is started according to the solution computed in the first step. On each level of the tree, the β most promising nodes are expanded, while all other nodes are pruned. The evaluation of a node is based on the earliest possible completion time of the corresponding operation based on the worker assignment.

For further details on the above algorithms and a more sophisticated improvement procedure, we refer to [10], where we also provide an extensive computational study that shows the heuristic approaches to be able to provide high-quality solutions within reasonable time. Here, we provide an additional

analysis regarding the priority rules in the above constructive procedure. Based on [11], where the performance of multiple priority rules for JSPs and FJSPs under different objective functions, including settings with setup times, is analyzed, we have implemented three priority rules, *shortest processing time* (SPT) incorporating *shortest setup time* (SS) (SPT + SS), *most work remaining* (MWKR), and a combination of the *flow due date* (FDD), the MWKR, and the SS priority rules (FDD/MWKR + SS). To analyze these priority rules, we performed computational tests on a PC with an Intel® Core™ i7-4770 CPU, running at 3.4 GHz, with 16 GB of RAM under a 64-bit version of Windows 8. We used IBM ILOG CPLEX in version 12.7 as a MIP solver.

Our random testbed is composed of 10 instance sets with differing numbers of jobs $|I|$, machines $|M|$, and workers $|W|$. Each set features 10 randomly generated instances, where the number of operations q_i, the number of eligible machines $|M_{i_j}|$, and the integer setup times $s^m_{i_j,k_l}$ are drawn from uniform distributions over the intervals given in Table 1. The generation of the processing times of the operations is as follows. We first draw auxiliary integer parameters p_{i_j} for all $i \in I$ and $i_j \in O_i$ from uniform distributions over the intervals given in Table 1. Based on these parameters, varying processing times over the corresponding eligible machines and workers are constructed. For details, we refer to [10]. With respect to the beam search approach, we set $\beta = 25$. Table 1 presents the computational results. It relates to calling CPLEX on the integrated MIP model presented in Sect. 2 and the constructive procedure (denoted by H) using the different priority rules. For each solution approach, the table includes the average objective function value of the best solutions returned by the respective algorithm (columns "C^{avg}_{max}"). Moreover, it includes information about the percentage of instances that were solved to optimality within a time limit of 3600 s (column "opt.") by CPLEX.

It can be seen that the heuristic approach clearly outperforms CPLEX for large instances. The priority rule (FDD/MWKR + SS) performs significantly better than the other priority rules. The maximum runtime of the constructive procedure that we encountered in our tests was about 8 s over all priority rules, so that this approach shows to be a reliable candidate for real-world usage.

Table 1 Computational results for the MIP model and the heuristic using different priority rules

Instance set							CPLEX		H_{SPT+SS}	H_{MWKR}	$H_{FDD/MWKR+SS}$
$\|I\|$	$\|M\|$	$\|W\|$	q_i	$\|M_{i_j}\|$	p_{i_j}	$s^m_{i_j,k_l}$	opt. [%]	C^{avg}_{max}	C^{avg}_{max}	C^{avg}_{max}	C^{avg}_{max}
4	2	2	[2, 3]	[2, 2]	[3, 6]	[1, 4]	100	30	36.2	35.8	32.8
5	3	3	[2, 3]	[2, 2]	[3, 6]	[1, 4]	100	27.1	37.5	32.8	30.4
6	3	3	[2, 3]	[2, 2]	[3, 6]	[1, 4]	60	30.4	38.3	38	34.8
7	4	3	[2, 3]	[2, 2]	[3, 6]	[1, 4]	10	32.9	39.4	38.1	35.5
8	4	3	[2, 3]	[2, 2]	[3, 6]	[1, 4]	–	37.4	42.3	40.8	39.4
10	5	4	[2, 3]	[2, 3]	[4, 8]	[1, 5]	–	48	53.5	50.4	47.4
15	10	8	[4, 6]	[3, 5]	[5, 10]	[1, 7]	–	–	101.9	91	83.9
20	10	8	[4, 6]	[3, 5]	[5, 10]	[1, 7]	–	–	125.3	115.8	107.1
30	20	16	[5, 7]	[5, 7]	[5, 15]	[1, 10]	–	–	178	135.7	131.4
40	20	16	[4, 6]	[5, 7]	[5, 15]	[1, 10]	–	–	184.9	147	134.5

Acknowledgements This work has been supported by the European Union and the state North Rhine-Westphalia through the European Fund for Regional Development (EFRD). It has been conducted as part of the project "EKPLO: Echtzeitnahes kollaboratives Planen und Optimieren" (EFRE-0800463).

References

1. Allahverdi, A.: The third comprehensive survey on scheduling problems with setup times/costs. Eur. J. Oper. Res. **246**(2), 345–378 (2015)
2. Allahverdi, A., Soroush, H.M.: The significance of reducing setup times/setup costs. Eur. J. Oper. Res. **187**(3), 978–984 (2008)
3. Allahverdi, A., Gupta, J.N.D., Aldowaisan, T.: A review of scheduling research involving setup considerations. Omega **27**(2), 219–239 (1999)
4. Allahverdi, A., Ng, C.T., Cheng, T.C.E., Kovalyov M.Y.: A survey of scheduling problems with setup times or costs. Eur. J. Oper. Res. **187**(3), 985–1032 (2008)
5. Błazewicz, J., Ecker, K.H., Pesch, E., Schmidt, G., Węglarz, J.: Handbook on Scheduling: From Theory to Applications. Springer, Berlin (2007)
6. Brucker, P., Schlie, R.: Job-shop scheduling with multi-purpose machines. Computing **45**(4), 369–375 (1990)
7. 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)
8. Giffler, B., Thompson, G.L.: Algorithms for solving production-scheduling problems. Oper. Res. **8**(4), 487–503 (1960)
9. Hurink, J., Jurisch, B., Thole, M.: Tabu search for the job-shop scheduling problem with multi-purpose machines. OR Spect. **15**(4), 205–215 (1994)
10. Kress, D., Müller, D., Nossack, J.: A worker constrained flexible job shop scheduling problem with sequence-dependent setup times. OR Spect. **41**(1), 179–217 (2019)
11. Sels, V., Gheysen, N., Vanhoucke, M.: A comparison of priority rules for the job shop scheduling problem under different flow time-and tardiness-related objective functions. Int. J. Prod. Res. **50**(15), 4255–4270 (2012)

Part XV
Simulation and Statistical Modelling

OR Control Towers: A Concept for Optimizing the Performance of Complex Adaptive Operating Systems

Joachim Block and Stefan Pickl

1 Introduction

The main challenges for managers are how to get an operating system under control, to keep it under control, and foremost to optimize its performance for sustainable success. An operating system is a configuration of resources that are combined in order to deliver goods or services to the environment [1]. In the operations research (OR) context we face very different operating systems ranging from pure technical systems (e.g. brokering systems) over socio-technical ones (e.g. supply chains or traffic control systems) up to systems consisting only of social entities (e.g. a workforce system). As systems raise in size and complexity well established optimization approaches based on prediction and direct control seem to reach its limitations.

For instance, in our modern world we recognize a transformation from sequenced supply chains to dynamic, interconnected supply systems [2], i.e. supply networks, resulting in a need for better solutions. Despite intense research in supply chain management, decision makers often struggle with the dynamics and complexity inherent to supply networks. One reason may be that prediction and control are limited in contrast to linear supply chains. To overcome this barrier the right balance between control and emergence has to be found [3].

J. Block (✉)
Universität der Bundeswehr München, Department of Computer Science, COMTESSA, Neubiberg, Germany
e-mail: archibald@computer.org

S. Pickl
Department of Computer Science, Universität der Bundeswehr München, Neubiberg, Germany

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_61

2 Controlling Complex Adaptive Operating Systems

Supply networks [4] and many other systems we encounter in OR can be classified as complex adaptive operating systems (CAOS). A CAOS is an operating system being composed of interacting, autonomous, and goal directed entities called agents [5]. Important properties of a CAOS are adaption and learning. Thereby, the system exhibits rather complex and hard to predict emergent performance patterns [6].

Definition 1 A **complex adaptive operating system (CAOS)** is a collection

$$S = (A, R, X, Y)$$

with

- A being the **set of agents**, i.e. the entities of the CAOS
- $R = A \times A$ being the **set of relations** between the agents
- X being the **input set** of the CAOS
- Y being the **output set** of the CAOS

An agent $a \in A$ interacts with other agents defined by the set of relations R, the interaction links. It senses external stimuli as input and sends output to other agents or to the environment either by action or reaction. The collective behavior of the agents determines the output of the CAOS and, hence, the system's performance.

The structure of the CAOS, i.e. the sets of agents and relations, evolves as time passes by. Agents may decide to leave the system by an explicit decision or be forced to leave by an external event. For instance, retailers in a supply network enter the network while existing ones vanish due to bankruptcy or other incidents. On the other hand, new agents may enter the system. The system's structure does not change continuously but at discrete time steps or events. Therefore, the configuration of a CAOS at time t can be defined by $S_t = (A_t, R_t, X, Y)$.

Every new agent changes the structure of the CAOS as it has to be integrated into the system. The same applies for the disintegration of an agent. However, structural changes of the CAOS are not limited to integration and disintegration of agents. In fact, changing relations between agents are an important property of CAOS. A production site may frequently change its contractors that are delivering raw material needed for the production of finished goods in order to minimize costs or to maximize quality.

Although changes in structure are initiated by single agents and are locally limited, they can spread through the whole system like a cascade as other entities react by adjusting their interaction links. This adaption mechanism is one main reason for the emergent behavior of a CAOS.

2.1 The OR Control Tower Concept

A CAOS exhibits robustness against disruptions up to a certain extend. By learning and adaption the agents aim to ensure their long term existence and, hence, the success of the system itself. If an agent senses a disruption it reacts by adopting the behavior and adjusting the interaction links. Therefore, the CAOS is able to compensate disruptions such as an agent becoming dysfunctional.

However, there are situations when a CAOS is not able to recover immediately from a disruption anymore [7]. In case of a natural disaster for instance, supply networks—although important factors in an effective disaster management—regularly break down. The performance of the supply network is dramatically limited until full recovery. This results at least in an economic harm. The earthquake in Japan 2011 and the following local nuclear meltdown has disrupted the global supply networks and caused a damage of estimated 235 billion US-dollars around the globe [8]. The inherent adaption and learning mechanisms of a CAOS often work by far too slow to prevent respectively respond to drastic system failures.

Not only the Japanese natural disaster calls for robust CAOS that are invulnerable against performance drops. Rather, a CAOS has to be resilient against any kind of system failures or even antifragile as Nassim Talib remarks [9]. In order to improve a CAOS for sustainable maximum performance under every circumstances we introduce the concept of an OR control tower.

A control tower is a dedicated element having full overview of a system. Furthermore, it has the ability to send commands to individual elements in order to control the system's behavior. An airport control tower for example observes the taxiways, runways, and the air traffic. It controls the ground and air traffic by giving detailed orders. The aim of the air control tower is to realize a smooth and safe airport operation.

Control towers are also beginning to be used in other OR domains. Advances in information technology enable us to implement control tower functions in supply chain management. However, the focus of these control towers is mainly to provide visibility and transparency as well as decision support for human operators [10]. Up to now, the existing control tower implementations are more information hubs, focusing on status information and events, than autonomous decision units designed to guarantee optimal performance of supply networks.

We extend the concept of control towers into two dimensions to form what we call an OR control tower. First of all, we add an autonomous acting component that steadily optimizes the performance of the CAOS without limiting the strengths of such a system, i.e. emergence and adaption. Second, we generalize the concept and do not focus on certain kinds of CAOS. Thereby, our concept may be applied to a variety of OR problems such as among others supply networks, workforce systems, and traffic control systems.

2.2 The Architecture of the OR Control Tower

Our concept adds an autonomous controlling and optimization entity to the CAOS: the OR control tower. The OR control tower and the controlled CAOS together form what we call a OR control tower system (Fig. 1). In contrast to the system's agents, the OR control tower has full overview of the CAOS and is able to communicate with all agents. Furthermore, it has the power to reconfigure the CAOS by adding and removing agents as well as adjusting relations. We propose a three computation component architecture consisting of an analyzing component, a predictive component, and an optimization component.

Definition 2 An **OR control tower** is a collection

$$C = (X_C, Y_C, ANA, PRE, OPT, S)$$

with

- X_C being the **input set**
- Y_C being the **output set**
- ANA being the **analyzing component**
- PRE being the **predictive component**
- OPT being the **optimization component**
- S being the **CAOS to be controlled**

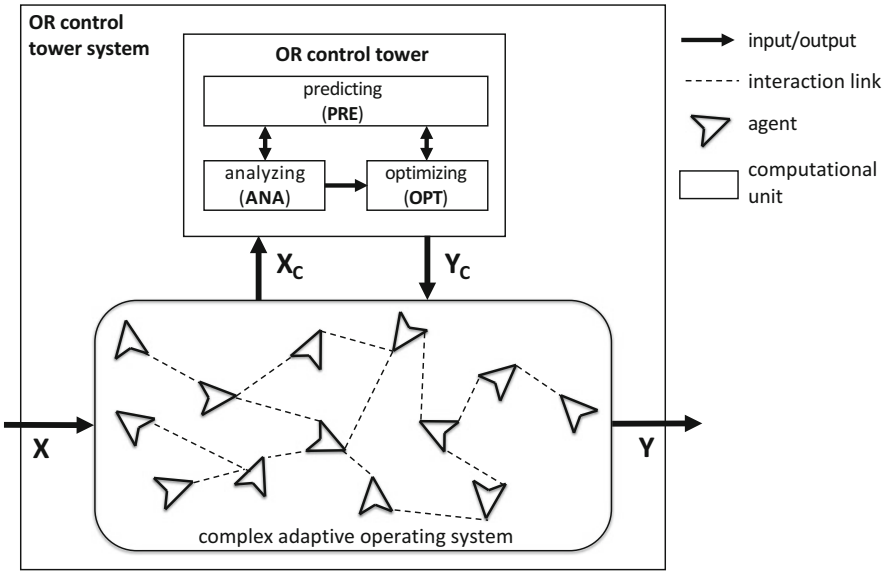


Fig. 1 The architecture of the OR control tower system

An OR control tower C has as input X_C not only the output of the CAOS. Furthermore, it also able to sense the actual state of every single entity of the CAOS. Hence, the OR control tower is linked to every agent. It steadily senses the state of the CAOS to identify if and how the system delivers the functionality it is designed for.

The input of the OR control tower is processed in its analyzing component (ANA). This component distinguishes between three different kinds of behavior of the CAOS (see e.g. [11]). It decides whether the functionality of the system

(a) can be provided in an optimal manner now and in the future,
(b) can be provided but not in an optimal way or not in the future, or
(c) cannot be provided at all.

However, although the actual state of the CAOS signals an optimal performance this does not necessarily mean that it will be true in the near future. In fact, the system can be at the edge of collapse. Therefore, it is not sufficient to analyze the current state of the CAOS but it is also necessary to consider the probable future evolution of the system. The predictive component of the OR control tower (PRE) delivers this kind of information. It is involved by the analyzing component for deciding whether to take a corrective action or not.

If a corrective action has to be taken, i.e. cases b) or c) above have been identified, the optimization component (OPT) is invoked. This component has implemented an optimization algorithm. It is among others able to identify the levering point of the CAOS. The optimization component has three alternatives to adjust the system's behavior. First of all, the structure of the system can be reconfigured. New agents can be introduced, exiting ones can be removed, and the interaction links can be adjusted. Second, the guiding policy for the whole CAOS can be replaced in favor of a better one. Third, one or more key entities can be directly influenced by the OR control tower in order to cause a cascading effect influencing the whole system; this is the levering point.

Direct commands and policies are output Y_C of the OR control tower directed to the system's entities. In order to perform a system reconfiguration the OR control tower has encoded the CAOS (similar to [12] or [13]).

It is important to not that the optimization component takes use of the predictive component as well. The main aim is not to optimize the behavior of the CAOS in the current context but to find an optimal and sustainable solution. Therefore, the optimization algorithm has to take into account the future evolution of the CAOS.

3 Performance Optimization

Our concept transforms a CAOS into an OR control tower system by introducing a dedicated entity that has not only full overview but also certain control over the system. However, this OR control tower is not meant to give detailed instructions to each agent of the CAOS which seems nearly impossible for complex OR systems.

Rather, it takes use of the emergent behavior caused by adaption and learning mechanisms. The OR control tower has the ability do reconfigure the CAOS or to adjust the guiding principles, i.e. the policies. Moreover, it can identify the levering point to force the system's behavior into an optimized direction. This means minimum effort with maximum impact without restricting the agents' autonomy to much.

The optimization algorithm implemented in the OR control tower could be for instance a simulation based approach. When an intervention has been identified the algorithm would simulate the further evolution of the CAOS under various control scenarios and choose the best scenario for action.

4 Conclusion

The OR control tower concept is a goal directed evolution of the control tower systems beginning to emerge in supply chain management. Extending the observation capability with an optimization algorithm that can take the probable future into account seems a logical step in designing and controlling CAOS. The OR control tower could help us in making better decisions when controlling complex OR systems. Furthermore, we could make our CAOS more stable and even resilient against disruptions by still drawing on the strengths of self adapting systems.

References

1. Wild, R.: Operations Management, 6th edn. Thomson, London (2002)
2. Mussomeli, A., Gish D., Laaper S.: The Rise of the Digital Supply Network. Deloitte University Press, Indianapolis (2016)
3. Choi, Th.Y., Dooley, K. J., Rungtusanatham, M.: Supply networks and complex adaptive systems: control versus emergence. J. Oper. Manag. **19**, 351–366 (2001)
4. Pathak, S. D., Day, J. M., Nair, A., Sawaya, W. J., Kristal, M. M.: Complexity and adaptivity in supply networks: building supply network theory using a complex adaptive systems perspective. Decis. Sci. **38**(4), 547–580 (2007)
5. Miller, J. H., Page S. E.: Complex Adaptive Systems – An Introduction to Computational Models of Social Life. Princeton University Press, Princeton (2007)
6. Holland, J. H.: Hidden Order – How Adaption Builds Complexity. Helix Books, New York (1995)
7. Snyder, L. V., Atan, Z., Peng, P., Rong, Y., Schmitt, A.J., Sinsoyal, B.: OR/MS models for supply chain disruptions: a review. IIE Trans. **48**(2), 89–109 (2015)
8. Park, Y., Hong, P., Roh, J.J.: Supply chain lessons from the catastrophic natural disaster in Japan. Bus. Horiz. **56**, 75–85 (2013)
9. Talib, N. N.: Antifragile – Things That Gain From Disaster. Random House, New York (2012)
10. Trzuskawska-Grzesińska, A.: Control towers in supply chain management – past and future. J. Eco. Manag. **27**(1), 114–133 (2017)

11. Güdemann, M., Nafz, F., Ortmeier, F., Seebach, H., Reif, W.: A specification and construction paradigm for organic computing systems. In: Proceedings of the 2nd IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO 2008), pp. 233–242
12. Block, J.: A hybrid modeling approach for incorporating behavioral issues into workforce planning. In: IEEE International Conference on Systems, Man, and Cybernetics, pp. 326–331 (2016)
13. Zeigler, B.P., Praehofer, H., Kim, T.G.: Theory of Modeling and Simulation – Integrating Discrete Event and Continuous Complex Dynamic Systems, 2nd edn. Academic Press, San Diego (2000)

Fighting Fair? Evaluating Negative Campaigning with an Agent-Based Simulation

Michelle D. Haurand and Christian Stummer

1 Motivation

Using the (hypothetical) war between humans and vampires to exemplify the emergence of dominant designs, we previously demonstrated that agent-based simulation is very well suited to study this complex process [1]. The results therein may be of interest for researchers and innovation managers alike, because they show that inferior technologies may establish themselves as dominant designs, as happened, for instance, with Blu-ray Disc winning over HD-DVD [2].

However, in examining how to prevent competing products from establishing themselves in a winner-take-all scenario, only measures promoting one's own technology were examined. Nowadays, humans might no longer have to fight against vampires, but there is a perhaps even more hostile "war" for dominance between technologies. Moreover, the evolving digital society encourages negative campaigning, which can be disseminated more easily via the internet [3]. Negative campaigning—the spreading of negative information about competitors—might in this special environment also receive a boost compared to traditional marketing, as negative news was found to be distributed farther and for a longer time than positive messages [4]. Moreover, the possibility of anonymity on the internet (which prevents backlash against the creators of negative campaigns), makes posting negative information (e.g., the infamous "fake news") about one's opponents online even more appealing, which is why the internet has grown in popularity as a medium for smear campaigns [5].

Although negative campaigning can be seen as a subform of comparative advertising [6], in a narrow sense, it is understood as advertising containing only

M. D. Haurand · C. Stummer (✉)
Bielefeld University, Bielefeld, Germany
e-mail: m.haurand@uni-bielefeld.de; christian.stummer@uni-bielefeld.de;
http://www.uni-bielefeld.de/(en)/wiwi/itm/

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_62

negative material about the competitor [7]. For the purpose of our research, we here use the second definition.

Negative campaigning might foster extreme views on competing products [8], which gives rise to the idea that it might also increase the probability of the emergence of a winner-take-all scenario. To the best of our knowledge, however, no studies on such an effect on product diffusion yet exist. While there is a large body of literature concerned with negative campaigning related to politics (see, e.g., [9–13]), and one study even explicitly explores the influence of negative political campaigning on a winner-take-all scenario [14], there is no equivalent literature in the fields of business studies or simulation. If there are studies concerned with spreading negative information about competing products or companies at all, there is always a focus on direct comparison between companies, that is, anonymity is not possible (see, e.g., [15–17]).

Our research contributions are thus twofold: Firstly, we explore the effect of negative campaigning on the emergence of a winner in a winner-take-all scenario. Secondly, we discuss the effectiveness of negative campaigning in contrast to traditional (positive) marketing measures.

The following Sect. 2 outlines how the vampire model mentioned above was improved and expanded to fit this scenario. In Sect. 3, we then present results from simulation experiments. The paper concludes with a summary and an outlook to promising avenues for further research in Sect. 4.

2 The Vampire Model

Our model is based on the description of a vampire economy by Farhat [18], in which humans manufacture stakes used for defending themselves against vampires, which feed on humans and possibly "turn" them. We improved and expanded this model to make it suitable for studying the emergence of a winner-take-all scenario, for example by including several different types of weapons (from most to least effective, these are stakes, mirrors, garlic, and silver bullets) [1]. Since then, we have again modified the model to include negative campaigning measures. Still, the humans represent potential customers who buy different technologies (i.e., weapons for fighting vampires), while the vampires can be seen as an unforgiving test of whether these weapons are efficient. If inferior weapons are chosen, the chance of winning the fight against a vampire decreases, which makes societies in which ineffective products win the market more prone to extinction. Vampires thus again represent the "problem" the technologies should solve.

An overview of our simulation steps, with differences to the ones by Farhat highlighted, can be found in Fig. 1. The white steps are directly taken from the model by Farhat, while the light gray steps are those we revised and/or extended and the dark gray steps are those we added (all of which are described in detail in [1]). Step 5 (marked with a thicker frame in the figure) was again modified for the current version of the model to include the possibility of negative campaigning.

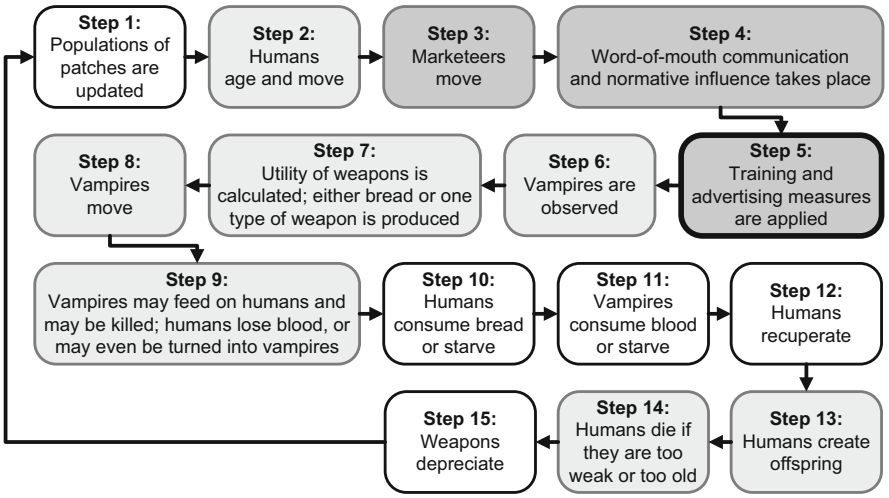


Fig. 1 Overview of simulation steps (adapted from [1])

The model passes through an initialization phase that is not included in Fig. 1. After this phase, in which all agents are randomly distributed across 1296 patches (i.e., 36×36 spatial units), the model recurrently runs through the 15 steps depicted in the figure:

Step 1: Populations (humans and vampires) of patches are recorded.

Step 2: Humans move to a more populated patch to be safer. Furthermore, aging was introduced in order to make the model more realistic, as only vampires can (theoretically) live forever and reproduce at the age of one.

Step 3: Marketeers travel the world to find a larger audience. Note that the original model by Farhat did not consider any marketing and therefore did not include marketeers.

Step 4: In the present model as well as in our last model, humans communicate about products and thereby influence each other's appraisal of the weapons' functionality (i.e., perceived effectivity). Furthermore, normative social influence, that is, peer pressure, takes place.

Step 5: Humans can learn how to wield the weapons and thus improve their individual effectivity in using it in a possible fight. Humans might also be affected by an advertising campaign executed by the marketeers. Such a campaign can aim at promoting a given weapon technology or it can be directed toward spreading negative information about some competing technology. During negative campaigns, either the most effective or the second most effective technology (i.e., stakes or mirrors) is attacked, as negative campaigning is typically used by inferior candidates against their superior opponents [9].

Step 6: Humans look out for vampires. This step was slightly modified to correct a shortcoming in Farhat's calculations.

Step 7: If humans recently observed a vampire, they produce a weapon. Otherwise, they produce bread to feed themselves. We added to this step by including a choice for a certain type of weapon, based on effectivity with the weapon, appraisal of its (potential) effectivity, and normative influence.

Step 8: Vampires move to a patch containing more blood and fewer dangerous weapons.

Step 9: Vampires hunt humans and possibly turn them. Note that turned humans appear in a random location in Farhat's model, while in the modified version, turned humans remain in the same place. In Farhat's model, vampires might die if too many stakes are present, while we made the outcome of the fight dependent on the individual effectivity of the humans with their weapon of choice.

Step 10: Humans eat or starve.

Step 11: Vampires eat or starve.

Step 12: Humans heal, that is, they recuperate blood.

Step 13: In Farhat's model, humans produce an identical hatchling that is sent to a random location, which greatly influences the model outcome. This is unrealistic and therefore was modified. In the current model, babies appear in the same location as their parents.

Step 14: Humans die from blood loss. In the current model, they may also die from old age.

Step 15: Weapons depreciate.

3 Results

Following [19], we set the threshold for a technology to be considered a dominant design at 50% market share in the final cycles (101 through 120) of a simulation run. For each scenario, 100 simulation runs were performed. In the baseline scenario, no measures were applied. In the remaining scenarios, we focused on an inferior technology, namely, garlic. First, we applied learning measures, in which (some) humans are trained to more effectively use some weapon, to boost this technology. Then, we did the same with traditional marketing measures by means of sending out marketeers, and finally we resorted to (anonymous) negative campaigning in two scenarios, making one of the two strongest competitors the aim of the negative campaigning per scenario. Table 1 shows the outcome of the simulation experiment, with the results for the originator of the negative campaigning (i.e., garlic) highlighted in bold.

It turns out that traditional (positive) marketing not only performs best but also increases the overall number of runs resulting in a dominant design. Negative campaigning against the most effective weapon (i.e., stakes) even slightly lowers the overall possibility of a dominant design appearing (i.e., increases the number of runs in which none of the technologies succeeded in becoming dominant from 48

Table 1 Number of runs in which dominant designs emerged

	Baseline	Training for garlic	Advertising for garlic	Negative campaigning against	
				Stakes	Mirrors
Stakes	28	16	17	13	39
Mirrors	19	15	7	26	7
Garlic	**5**	**23**	**47**	**10**	**10**
Bullets	0	0	0	1	0
None	48	46	29	50	44

to 50); doing the same against mirrors (the second most effective weapon) results at least in a slight increase of the number of instances with dominant designs being established (from 52 to 56).

While all measures improve the chance of garlic becoming the dominant weapon design, traditional marketing (i.e., advertising) is the most effective in our simulation experiments, followed by learning (i.e., training) measures, and then negative campaigning. Negative campaigning against stakes or mirrors mostly benefit the second best or best technology, respectively.

4 Conclusion

This work investigated the effectiveness of negative campaigning, as opposed to measures focusing on promoting one's own technology, that is, advertising or provision of practicing/training opportunities. Our application case of the competition between different weapon technologies for fighting vampires shows that even though negative campaigning might increase the chances of inferior technologies becoming the dominant design, management should rather resort to traditional (positive) marketing measures.

While this, of course, has been a hypothetical case, it still may serve as an illustrative example for demonstrating the potential of agent-based simulation for studying the effects of negative campaigning. Further research might be directed toward the investigation of countermeasures against negative campaigning. Moreover, it would be interesting to shed light on the effect of the timing for introducing some technology onto the market, which supposedly has played a role in the format war between Blu-ray Disc and HD-DVD mentioned in the motivation section of this paper.

References

1. Haurand, M.D., Stummer, C.: Stakes or garlic? Studying the emergence of dominant designs through an agent-based model of a vampire economy. Cent. Eur. J. Oper. Res. **26**, 373–394 (2018)

2. Brem, A, Nylund, P.A., Schuster, G.: Innovation and de facto standardization: the influence of dominant design on innovative performance, radical innovation, and process innovation. Technovation **50–51**, 79–88 (2016)
3. Mattes, K., Redlawsk, D.P.: The positive case for negative campaigning. University of Chicago Press, Chicago (2015)
4. Hornik, J., Satchi, R.S., Cesareo, L., Pastore, A.: Information dissemination via electronic word-of-mouth: good news travels fast, bad news travels faster! Comput. Hum. Behav. **45**, 273–280 (2015)
5. Mark, D.: Going dirty: the art of negative campaigning. Rowman & Littlefield, Lanham (2009)
6. Merritt, S.: Negative political advertising: some empirical findings. J. Advert. **13**, 27–38 (1984)
7. Hill, R.P.: An exploration of voter responses to political advertisements. J. Advert. **18**, 14–22 (1989)
8. James, K.E., Hensel, P.J.: Negative advertising: the malicious strain of comparative advertising. J. Advert. **20**, 53–69 (1991)
9. Auter, Z.J., Fine, J.A.: Negative campaigning in the social media age: attack advertising on Facebook. Polit. Behav. **38**, 999–1020 (2016)
10. Dolezal, M., Ennser-Jedenastik, L., Müller, W.C.: Negative campaigning and the logic of retaliation in multiparty competition. Int. J. Press/Polit. **21**, 253–272 (2016)
11. Dolezal, M., Ennser-Jedenastik, L., Müller, W.C.: Who will attack the competitors? How political parties resolve strategic and collective action dilemmas in negative campaigning. Party Polit. **23**, 666–679 (2017)
12. Hopp, T., Vargo, C.J.: Does negative campaign advertising stimulate uncivil communication on social media? Measuring audience response using big data. Comput. Hum. Behav. **68**, 368–377 (2017)
13. Song, H., Nyhuis, D., Boomgaarden, H.: A network model of negative campaigning: the structure and determinants of negative campaigning in multiparty systems. Commun. Res. 1–22 (2017)
14. Hopkin, J., Alexander Shaw, K.: Organized combat or structural advantage? The politics of inequality and the winner-take-all economy in the United Kingdom. Polit. Soc. **44**, 345–371 (2016)
15. Yucel-Aybat, O., Kramer, T.: Comparative advertisements and schadenfreude: when and why others' unfortunate choices make us happy. Mark. Lett. **28**, 579–589 (2017)
16. Yucel-Aybat, O., Kramer, T.: The impact of competitiveness on consumer responses to comparative advertisements. J. Advert. **47**, 198–212 (2018)
17. Grewal, D., Kavanoor, S., Fern, E.F., Costley, C., Barnes, J.: Comparative versus noncomparative advertising: a meta-analysis. J. Mark. **61**, 1–15 (1997)
18. Farhat, D.: The Economics of Vampires: An Agent-Based Perspective. Working paper no. 1301, University of Otago, Department of Economics, Dunedin (2013)
19. Anderson, P., Tushman, M.L.: Technological discontinuities and dominant designs: a cyclical model of technological change. Admin. Sci. Quart. **35**, 604–633 (1990)

A Variational Inequality Approach to Optimal Control Problems with Joint Constraints

Zhengyu Wang and Stefan Pickl

1 Problem Formulation

The optimal control problems (OCP) are ubiquitous in engineering and economics, among which those with mixed algebraic state-control constraints are the most challenging [4, 6]. Given matrices $A, E, K \in R^{n\times n}$, $B, Q \in R^{m\times n}$, $M \in R^{m\times m}$, $C \in R^{\ell\times n}$ and $D \in R^{\ell\times m}$, given functions $r, p : [0, T] \to R^n$, $q : [0, T] \to R^m$ and $g : [0, T] \to R^\ell$, given vectors $y^0, c \in R^n$. We study the OCP

$$\begin{aligned} \min\ & \frac{1}{2}y(T)^T Ey(T) + c^T y(T) + \int_0^T \varphi(t, y(t), u(t))dt \\ s.t.\ & \dot{y}(t) = Ay + Bu + r(t), \quad y(0) = y^0 \\ & Cy(t) + Du(t) + g(t) \geq 0, \end{aligned} \tag{1}$$

where

$$\varphi(t, y, u) = \frac{1}{2}u^T Mu + \frac{1}{2}y^T Ky + u^T Qy + u^T q(t) + y^T p(t).$$

Z. Wang
Department of Mathematics, Nanjing University, Nanjing Shi, China
e-mail: zywang@nju.edu.cn

S. Pickl (✉)
Department of Computer Science, Universität der Bundeswehr München, Neubiberg, Germany
e-mail: stefan.pickl@unibw.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_63

By using the Pontryagin's principle, we reformulate the OCP as a hybrid system:

$$\begin{cases} \dot{v}(t) = -\nabla_y H(t, v(t), y(t), u(t)) \\ \dot{y}(t) = \quad \nabla_v H(t, v(t), y(t), u(t)) \\ u(t) \in \arg\min_{z \in U(t, y(t))} H(t, v(t), y(t), z) \\ y(0) = y^0 \ \text{ and } \ v(T) = \nabla_y \psi(y(T)), \end{cases} \tag{2}$$

where $U(t, y)$ is a set-valued mapping and $H(t, v, y, u)$ denotes the Hamiltonian

$$\begin{aligned} U(t, y) \quad &= \{u \in R^m \mid Cy + Du + f(t) \geq 0\} \\ H(t, v, y, u) &= \tfrac{1}{2} u^T M u + \tfrac{1}{2} y^T K y + u^T Q y \\ &\quad + u^T q(t) + y^T p(t) + v^T (Ay + Bu + r(t)). \end{aligned}$$

The optimality of the quasi-optimization problem in (2) can be formulated as a *quasi variational inequality* (QVI): Find $u \in U(t, y)$ such that

$$(z - u)^T \nabla_u H(t, v, y, u) \ \geq \ 0 \quad (\forall z \in U(t, y)), \tag{3}$$

where the solution set is denoted by $\mathrm{SOL}(U(t, y), \nabla_u H(t, v, y, \cdot))$. Replacing the optimization by the (3), we are given a *differential quasi variational inequality*:

$$\begin{cases} \dot{y}(t) = Ay(t) + Bu(t) + r(t) \\ \dot{v}(t) = -Ky(t) - A^T v(t) - Q^T u(t) - p(t) \\ u(t) \in \mathrm{SOL}\left(U(t, y(t)), Qy(t) + B^T v(t) + q(t) + M(\cdot)\right) \\ y(0) = y^0 \ \text{ and } \ v(T) = Ey(T) + c. \end{cases} \tag{4}$$

The DQVI has been touched little, it reduces into the *differential variational inequality* (DVI) if the constraint is only on the control [7]. The time stepping scheme is the most popular method for the DVIs [2, 4], it was shown having a 1-order convergence in the best case where the involved VI is uniquely solvable [1], which is generally not fulfilled for the OCP (1).

In this paper, by reformulating the OCP into a VI, we establish the solvability of the OCP and propose a regularized Galerkin method, it approximates the least norm solution of the OCP by solving a strongly monotone finite-dimensional VI.

2 Variational Inequality Reformulation

Denote by $L^2(0, T; R^\ell)$ the Hilbert space of ℓ-dimensional vector-valued functions that are square integrable. Introduce a linear operator $\mathcal{T}$ on $L^2(0, T; R^n)$:

$$(\mathcal{T} y)(t) = \int_0^t e^{(t-s)A} y(s) ds. \tag{5}$$

Obviously, by the constant variation formula, we have:

$$y(t) = e^{tA}y^0 + (\mathcal{T}r)(t) + (\mathcal{T}Bu)(t). \tag{6}$$

This helps removing the state variable from the joint constraint, which is reformulated as a constraint purely on the control

$$\Omega = \{u \in L^2(0, T; R^m) : (\mathcal{L}_0 u + \tilde{f})(t) \geq 0, \ a.e. \ in \ [0, T]\}, \tag{7}$$

where $\mathcal{L}_0$ is a bounded operator on $L^2(0, T; R^m)$ and $\tilde{f} \in L^2(0, T; R^m)$, given by:

$$\begin{aligned}(\mathcal{L}_0 u)(t) &= C\mathcal{T}(Bu)(t) + Du(t)\\ \tilde{f}(t) &= Ce^{tA}y^0 + C\mathcal{T}(r)(t) + f(t).\end{aligned}$$

In the remaining part we omit "a.e." in the (7) for an easy presentation.

Let (4) have a weak solution (y, v, u) with $u \in L^2(0, T; R^m)$. Again applying the constant variation formula to the adjoint equation of (4), we have

$$v(t) = e^{(T-t)A^T}v(T) + \mathcal{T}^*q(t) + \mathcal{T}^*(Ky)(t) + \mathcal{T}^*(Q^T u)(t), \tag{8}$$

where $\mathcal{T}^*$ denotes the adjoint of the $\mathcal{T}$. Plugging (6) and (8) into the QVI involved in the (4) we obtain

$$\Phi(u)(t) \ = \ Qy(t) + B^T v(t) + p(t) + Mu(t) = (\mathcal{L}u)(t) + b(t), \tag{9}$$

where $\mathcal{L}u = Mu + \mathcal{L}_1 u + \mathcal{L}_2 u + \mathcal{L}_3 u$,

$$\begin{aligned}\mathcal{L}_1 u(t) &= \int_0^T (e^{(T-t)A}B)^T E e^{(T-s)A}Bu(s),\\ \mathcal{L}_2 u(t) &= \int_t^T ds \int_0^s (e^{(s-t)A}B)^T K e^{(s-\tau)A}Bu(\tau)d\tau,\\ \mathcal{L}_3 u(t) &= \int_0^t Qe^{(t-s)A}Bu(s)ds + \int_t^T (Qe^{(s-t)A}B)^T u(s)ds,\end{aligned}$$

and

$$\begin{aligned}b(t) = p(t) &+ Qe^{tA}y^0 + (e^{(T-t)A}B)^T(c + Ee^{TA}y^0)\\ &+ \int_0^T (e^{(T-t)A}B)^T E e^{(T-s)A}f(s)ds + \int_0^t Qe^{(t-s)A}f(s)ds\\ &+ \int_t^T (e^{(s-t)A}B)^T \left[q(s) + Ke^{sA}y^0 + \int_0^s Ke^{(s-\tau)A}f(\tau)d\tau\right].\end{aligned}$$

Obviously, if (y, u) solves the OCP (1) with $u \in L^2(0, T; R^m)$, then $u \in \Omega$ and

$$\langle w - u, \Phi(u)\rangle_{L^2} = \int_0^T [w(t) - u(t)]^T \Phi(u(t))dt \ \geq \ 0. \quad (\forall w \in \Omega) \tag{10}$$

This is just a variational inequality posed in the $L^2(0, T; R^m)$, which we denote by $VI(\Omega, \Phi)$. Throughout this paper we suppose

(A) the matrices E and

$$\Xi = \begin{pmatrix} M & Q \\ Q^T & K \end{pmatrix}$$

are symmetric positive semi-definite;
(B) the functions r, f, p, q involved in the OCP (1) are square integrable;
(C) the Ω and the interior of its barrier cone Ω_b are both nonempty, where

$$\Omega_b = \{v \in L^2(0, T; R^m) : \sup_{u\in\Omega} \langle v, u\rangle_{L^2} < \infty\};$$

(D) $Mu + Qy = 0, Q^T u + Ky = 0, Cy + Du \geq 0$ and $y \in range(G)$ imply $u = 0$, where $G = G(A, B)$ denotes the controllability matrix of the controlled system of (1):

$$G(A, B) = (B, AB, \cdots, A^{n-1}B) \ \in \ R^{n\times(mn)}.$$

Remark 1 In [4] Han et al. reformulated the OCP into a complementarity system (a special DVI), they showed the convergence of the time-stepping method for the complementarity system and showed the solvability of the OCP by using the convergence. Some conditions used in [4] are somehow technical, and more restrictive than ours.

Below we present only the theoretical results.

3 Solvability of OCP

First of all, we mention that a solution of the $VI(\Omega, \Phi)$ yields an optimal solution of the OCP (1), which justifies our VI reformulation (10). Precisely we have:

Theorem 1 *If (y^*, u^*) is an optimal solution of the (1) in the weak sense with $u^* \in L^2(0, T; R^m)$, then u^* is a solution of the $VI(\Omega, \Phi)$. If u^* is a solution of the $VI(\Omega, \Phi)$ and y^* is given by the (6), then (y^*, u^*) is an optimal solution of the OCP (1) in the weak sense.*

Subsequently we treat the monotone and coercive properties of the Φ.

Theorem 2 *If the matrix E is positive semi-definite, then $\mathcal{L}_1$ is monotone. If the matrix Ξ is positive semi-definite, then $M + \mathcal{L}_1 + \mathcal{L}_1$ is monotone.*

Theorem 3 *The mapping Φ is coercive: there exists a $u^0 \in \Omega$ such that*

$$\frac{\langle \Phi(u), u - u^0 \rangle_{L^2}}{\|u\|_{L^2}} \to +\infty. \quad (as\ \|u\|_{L^2} \to \infty\ with\ u \in \Omega)$$

We conclude this section by presenting the solvability result for the OCP (1).

Theorem 4 *The solution set of the $VI(\Omega, \Phi)$ is bounded, the set of the optimal solutions (y, u) of the OCP (1) with $u \in L^2(0, T; R^m)$ is bounded.*

4 Regularized Galerkin Approximation

Let X^h be a finite-dimensional subspace of $L^2(0, T; R^m)$ parameterized by h, for example, the stepsize of the subdivision of the $[0, T]$. Let P_h be the orthogonal projection onto the X^h, it is linear bounded symmetric and fulfills

$$\|I - P_h\|_{L^2} = C(h) \to 0. \quad (h \downarrow 0)$$

Let $\lambda \geq 0$. By the regularized Galerkin approximation we mean approximating the infinite-dimensional $VI(\Omega, \Phi)$ by the finite-dimensional $VI(\Omega^h, \Phi^{h,\lambda})$, where

$$\Omega^h := \Omega \cap X^h, \qquad \Phi^{h,\lambda}(u) = P_h \Phi(u) + \lambda u.$$

The set Ω^h is nonempty for a suitable subspace X^h, it is closed and convex. The Ω^h is an approximation of the Ω in the sense that the metric projection $v^h = P_{\Omega^h} v$ of any $v \in \Omega$ is strongly convergent to v as $h \to 0$, this is because

$$\|v - v^h\|_{L^2} \ \leq\ \|v - P_h v\|_{L^2} \ \leq\ C(h)\|v\|_{L^2}. \tag{11}$$

We establish at first the monotonicity of the $\Phi^{h,\lambda}$.

Theorem 5 *Under the condition (A), the $\Phi^{h,\lambda}$ is monotone when $\lambda = 0$, and is strongly monotone when $\lambda > 0$.*

For $\lambda > 0$, the $\Phi^{h,\lambda}$ is strongly monotone, and therefore the $VI(\Omega^h, \Phi^{h,\lambda})$ has the unique solution, which can be efficiently computed by the existing software [3]. We show that its solution is uniformly bounded in λ.

Theorem 6 *Let $\bar{h} > 0$ be given, take $\lambda = \lambda(h) \to 0$ as $h \to 0$, let u^h be a solution of the $VI(\Omega^h, \Phi^{h,\lambda})$ for $0 < h \leq \bar{h}$. Then the set $\{u^h\}_{0 \leq h \leq \bar{h}}$ is bounded.*

Below we show the convergence of the solutions given by the regularized Galerkin approximation.

Theorem 7 *Take $\lambda = \lambda(h) \to 0$ as $h \to 0$, and denote by u^h an arbitrary but fixed solution of the $VI(\Omega^{h_k}, \Phi^{h_k,\lambda_k})$. Then there is a sequence $\{h_k\} \downarrow 0$ such that the sequence $\{u^{h_k}\}$ is weakly convergent to a solution u of the $VI(\Omega, \Phi)$, moreover the sequence of the state $\{y^{h_k}\}$ is uniformly convergent to y, where (y, u) is an optimal solution of the OCP (1), and where*

$$y^{h_k}(t) \;=\; e^{tA}y^0 + \int_0^t e^{(t-s)A}r(s)ds + \int_0^t e^{(t-s)A}Bu^{h_k}(s)ds.$$

Remember that the solution set of the $VI(\Omega, \Phi)$ is nonempty closed and convex, so it has a least norm element, which is the metric projection of 0 onto the Ω. The least norm solution is of physical significance. Finally, we show that such a solution can be approximated by the regularized Galerkin scheme.

Theorem 8 *Take $\lambda = \lambda(h) > 0$ such that*

$$\lim_{h\downarrow 0} \frac{C(h)}{\lambda(h)} = 0, \qquad C(h) = \|I - P_h\|_{L^2}.$$

Denote by u^h the (unique) solution of the $VI(\Omega^h, \Phi^{h,\lambda})$. Then the $\{u^h\}$ has a subsequence that is weakly convergent to the least norm solution u of the $VI(\Omega, \Phi)$.

In a forthcoming contribution we specify the proofs and extend the new regularized Galerkin method. Furthermore, we apply this method to real world examples, especially in the context of biosystems motivated by Krabs and Pickl [5].

Acknowledgements The work was supported by the National Natural Science Foundation of China (No. 11871268).

References

1. Chen, X., Wang, Z.: Error bounds for a differential linear variational inequality. IMA J. Numer. Anal. **32**, 957–982 (2011)
2. Chen, X., Wang, Z.: Convergence of regularized time-stepping methods for differential variational inequalities. SIAM J. Optim. **23**, 1647–1671 (2013)
3. Dirkse, S., Ferris, M., Munson, T.: The PATH Solver, http://pages.cs.wisc.edu/~ferris/path.html, last retrieved in Sept. 2017
4. Han, L., Camlibel, M.K., Pang, J.-S., Heemels, W.P.M.H.: A unified numerical scheme for linear-quadratic optimal control problems with joint control and state constraints. Optim. Meth. Softw. **27**, 761–799 (2012)
5. Krabs, W., Pickl, S.W.: Modeling, Analysis and Optimization of Biosystems. Springer, Berlin (2007)
6. Kuhn, M., Wrzaczek, S., Prskawetz, A., Feichtinger, G.: Optimal choice of health and retirement in a life-cycle model. J. Econom. Theory **158**, 186–212 (2015)
7. Pang, J.-S., Stewart, D.E.: Differential variational inequalities. Math. Program. **113**, 345–424 (2008)

Part XVI
Software Applications and Modelling Systems

Adaptive Algorithmic Behavior for Solving Mixed Integer Programs Using Bandit Algorithms

Gregor Hendel, Matthias Miltenberger, and Jakob Witzig

1 Introduction

Most modern MIP solvers employ the LP-based branch-and-bound method. At each node of the search tree, a relaxation of the problem is solved by the dual Simplex algorithm to obtain both improving solutions and lower bounds on the problem. Many primal heuristics have been proposed (see [4] for an overview and further references) to aid the branch-and-bound procedure in finding good feasible solutions quickly. Two important subclasses of primal heuristics for MIP are large neighborhood search (LNS) and diving heuristics (Sects. 3.2 and 3.3). In total, `SCIP` [8] in version 5.0 features 12 diving heuristics, 10 pure LNS heuristics, further LNS heuristics for MINLP as well as certain primal heuristics that solve auxiliary problems in special cases. As there is no theoretical reason why certain heuristics work better for a MIP instance, tuning them individually can be a tedious task. Similarly, a large amount of the total solving time is spent during LP re-optimizations, so the choice of an efficient pricing method can be crucial (Sect. 3.1).

In all three cases, it is desirable to "learn" the best performing algorithms during the solving process. To this end, we investigate computational benefits of adaptive algorithmic behavior, governed by algorithms for the multi-armed bandit problem (Sect. 2). Such bandit algorithms try to balance their selection carefully between exploration among the available set of algorithms and exploitation of the best performing ones. We discuss suitable reward functions to measure the success of an algorithm for each class, and present promising results in a computational study.

G. Hendel (✉) · M. Miltenberger · J. Witzig
Zuse Institute Berlin, Berlin, Germany
e-mail: hendel@zib.de; miltenberger@zib.de; witzig@zib.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_64

2 Bandit Selection Strategies

The selection among a set of actions under uncertain payoffs appears as *multi-armed bandit problem* (MAB)[5]. In each *round* $t > 0$, a player selects an *action* a_t out of a set of actions $\mathcal{A}$. In turn for playing a_t, the player observes a *reward* $r_{a_t,t} \in [0, 1]$ with the goal to maximize the *total reward* $\sum_t r_{a_t,t}$. We call algorithms for MAB *selection strategies*. MAB distinguishes two scenarios. In the *stochastic scenario*, rewards are drawn from a reward distribution R_a that is independent of t. As the expected rewards are not known to the player beforehand, a good selection strategy carefully balances exploration between the actions and exploitation of the best single action. A common strategy to address stochastic scenarios is based on *upper confidence bounds* (`UCB`). Let $T_{a,t}$ denote the number of times that action $a \in \mathcal{A}$ has been selected until round t, and let $\bar{r}_{a,t}$ denote the mean reward of action a after a has been selected once ($T_{a,t} > 0$). After playing every action once, `UCB` selects the action that maximizes an upper confidence bound

$$a_t := \operatorname*{argmax}_{a \in \mathcal{A}} \left\{ \bar{r}_{a,t-1} + \sqrt{\alpha \cdot \frac{\ln(1+t)}{T_{a,t-1}}} \right\}$$

on the expected reward. The confidence band around a mean reward increases with every round in which an action is not played, forcing the selection of inferior actions from time to time. Its width is controlled by a parameter $\alpha \geq 0$. Note that a `UCB` strategy acts entirely greedy for $\alpha = 0$.

The second MAB scenario is the *adversarial scenario*, in which the rewards are picked by an opponent aiming at maximizing the player's total regret. Selection strategies for this scenario usually involve weighted sampling from an incrementally updated probability distribution $p_{a,t} = \mathbb{P}(a_t = a)$, starting from a uniform distribution $p_{a,1} = \frac{1}{|A|}$. A special variant of weighted sampling is the `Exp.3` selection strategy [5], which scales an observed reward with the current selection probability of action a_t. Therefore, if an action with small selection probability yields a high reward, this action will have a much higher chance to be selected in subsequent rounds.

3 Adaptive Algorithmic Behavior

We give three examples of branch-and-bound solving components suitable for an adaptive selection strategy. At the end of each section, we show the individual impact on the academic MIP solver `SCIP` and LP solver `SoPlex` [8]. All experiments are based on a pre-release version of `SCIP` 6.0 and `SoPlex` 3.1.1. The experiments have been performed on a test set MMMC of 496 instances combining the benchmark sets MIPLIB 3, MIPLIB 2003, MIPLIB 2010, and COR@L (see [10]

and [6]). Each experiment has been conducted on a cluster with identical machines to ensure comparable running time measurements.

3.1 Pricing for the Dual Simplex Algorithm

The dual Simplex algorithm is one of the most important techniques for LP problems and key for the LP-based branch-and-bound approach. Among the few algorithmic choices within the Simplex algorithm, one is the determination of the direction to search for a new basic solution, called *pricing step*. In this paper, we consider three well-known and practically proven methods called *devex pricing* [9], *steepest edge pricing*, and *quick start steepest edge* [7]. All methods try to select a direction that is steepest in regard to the dual objective improvement, thereby balancing accuracy of the decision and computational overhead per iteration. Devex pricing requires the least work per iteration but may lead to a higher number of total iterations. On the other hand, steepest edge computes accurate improvement measures that often lead to a considerable smaller iteration count, and an initialization step that can be expensive to compute, depending on the starting basis. While this is less relevant for pure LP solving, in the branch-and-bound context many LP re-optimizations are performed that start from an advanced basis. Here, quick start steepest edge sacrifices accuracy for a faster initialization. We refer to the literature for an in-depth description of these pricing techniques.

In our computational study we compare average LP throughput (LPs per seconds) and running time of three fixed pricers and three bandit selection variants after the root node has been processed, see Table 1. The selection strategies have to select from the set of available actions $\mathcal{A} = \{\texttt{devex},\texttt{qsteep},\texttt{steep}\}$. As the `UCB` strategy (cf. Sect. 2), requires a reward within the interval $[0, 1]$, we scale the measured running time $\tau_{a,t}$ of pricer a at time step t as $\frac{1}{1+\tau_{a,t}/\bar{\tau}_t}$, where $\bar{\tau}_t$ denotes the average running time of all LP resolves so far, independently of the selected pricer. A value of $\alpha = 2$ is used for `UCB`. We also test a `greedy` strategy that always selects the pricer with minimum modified average running

Table 1 Results for LP pricers. Columns: shifted geom. mean LP throughput (**LPthpt**, shift: 1), time in seconds (**time**, shift: 1), and respective quotients (**LPthpt$_\mathbf{Q}$**, **time$_\mathbf{Q}$**)

Pricer	Solved	LPthpt	LPthpt$_\mathbf{Q}$	Time	Time$_\mathbf{Q}$
`devex`	64	74.24	1.000	91.82	1.000
`steep`	65	62.66	*0.844*	99.41	*1.083*
`qsteep`	60	58.00	*0.781*	101.13	*1.101*
`UCB`	63	79.02	**1.064**	92.80	1.011
`weighted`	65	71.93	0.969	93.95	1.023
`greedy`	65	85.06	**1.146**	89.11	0.970

105 instances, 4 LP seeds, 900 s time limit
Bold values indicate a relative improvement by at least 0.05 (e.g., 5%), whereas relative deteriorations by the same amount are italicized

time $\bar{\tau}^{\sigma}_{a,t} = \sum_{t':a_t=a} \tau_{a,t'}/(T_{a,t} + \sigma_a)$, using shift values of $\sigma_a = 100$ for `devex` and $\sigma_a = 50$ for the other two. The favorite pricer `devex` is also kept if its LP iteration count stays below 20 on average. The use of the shift values encourages more exploration among the available pricers at the beginning. As a last variant (`weighted`), we use the modified means of the `greedy` selection method as input for a weighted sampling. We initialize the sampling probabilities as $p_{a,t} \propto (\bar{\tau}^{\sigma}_{a,t} + 10^{-4})^{-1}$. Here, the symbol $\propto$ expresses "proportional to", that means up to a scaling constant. Among the fixed pricers, `devex` is clearly the one with the highest LP throughput. The throughput can be increased by 6% when using `UCB`, and even 14% when using `greedy`. In contrast, `weighted` does not yield an improved LP throughput. While the `greedy` strategy even yields a 3% time improvement, the positive result of `UCB` for the LP throughput is still too marginal to make `SCIP` consistently solve problems faster on average.

3.2 Large Neighborhood Search Heuristics

The first class class of algorithms that we studied in the context of adaptive behavior are the Large Neighborhood Search (LNS) heuristics. Briefly, an LNS heuristic solves an auxiliary MIP under strict working limits, which is derived from the original MIP by fixing variables, adding constraints, and/or changing the objective function. In total, `SCIP` features 10 LNS heuristics, eight of which we integrated into a framework called Adaptive Large Neighborhood Search (ALNS). ALNS adapts the selection of the next LNS heuristic that should be executed based on the average reward observed so far. Besides the adaptive selection procedure, ALNS features more techniques such as a dynamic target fixing rate and a generic variable fixing procedure. ALNS has been first released with `SCIP` 5.0, and further improved for `SCIP` 6.0. Its reward function has been designed to prefer LNS heuristics that find improving solutions with a small computational effort. To this end, it convexly combines a simple indicator function whether an improving solution has been found with the obtained gap that has been closed. The obtained score is then scaled by the involved effort, as a function of the fixing rate and the number of nodes spent inside the sub-MIP.

On the test set used for the present work, the individual parameters for the selection strategies have first been optimized by a separate simulation procedure. With those parameters, ALNS has been called on a total of 445 problem instances. With an `Exp.3` strategy, it could solve two more instances than the `SCIP` default, and yielded a speed-up of 2.3%, and even 4.6% on the subset of MIPLIB 2010 benchmark instances, solving one additional instance. Details about the simulation procedure as well as the results will be presented in an own technical report about ALNS that is currently under preparation.

3.3 Diving Heuristics

Another class of heuristics are *diving heuristics*. Starting from a fractional LP solution, diving heuristics explore an auxiliary search tree in a depth-first fashion. The branching rules used in diving heuristics usually tend towards feasibility. In contrast to that, branching rules of the main search process, e.g., reliability branching [2], focus on a good subdivision of the problem. For an overview of the diving heuristics available in `SCIP`, we refer to [3]. In `SCIP`, diving heuristics also provide useful search information. For example, domain propagation is applied after rounding variables, to reduce variable domains or even detect infeasibility. The latter can be analyzed by conflict analysis techniques, e.g., [1, 11], to derive additional global information.

For our computational experiments, we have extended `SCIP` by a new primal heuristic plugin that selects one out of nine available diving heuristics at each call. A weighted sampling is used as selection strategy, where the sample weight of a diving heuristic (action) a is computed as

$$p_{a,t} \propto \left(\frac{\sum_{t'} b_{a,t'} + 100}{\sum_{t'} c_{a,t'} + 100} + 10^{-4} \right)^{-1}$$

with $b_{a,t'}$ denoting the number of backtracks performed by a at round t', and $c_{a,t'}$ denoting the number of conflict constraints generated. Both values are 0 if a has not been selected in round t'.

Table 2 compares the performance of `SCIP` in its standard configuration (`default`) and with adaptive diving selection (`adaptivediving`) on the MMMC test set with a time limit of one hour. An instance is called "solved" only if it has been solved consistently with each of three tested random seeds. Using `adaptivediving`, `SCIP` could solve seven more instances consistently within the time limit. On non-trivial instances where at least one configuration needs 10 or more seconds, it leads to speed up of almost 8%. On the state-of-the-art benchmark

Table 2 Aggregated results for adaptive diving over three random seeds

	Instances	`Default`			`Adaptivediving`		
		Solved	Nodes	Time	Solved	Nodes_Q	Time_Q
All	491	320	2550	152	327	**0.938**	0.958
Affected	284	274	1120	46	281	**0.939**	**0.945**
[10,tilim]	245	210	2693	158	217	**0.899**	**0.922**
[100,tilim]	144	109	5821	526	116	**0.904**	**0.909**
MIPLIB 2010	86	67	5064	301	70	**0.930**	0.969

Columns: shifted geom. mean of generated nodes (**nodes**, shift: 100), solving time in seconds (**time**, shift: 1), and respective quotients ($\textbf{nodes}_\text{Q}$ and $\textbf{time}_\text{Q}$)
Bold values indicate a relative improvement by at least 0.05 (e.g., 5%)

set MIPLIB 2010, `adaptivediving` leads to a slight performance improvement by 3% and three more solved instances.

4 Conclusion

We have proposed adaptive control mechanisms for three algorithm classes within `SCIP`. For each class, we introduce a suitable reward function to rank the different algorithms. The adaptive LP pricing intuitively has the nature of a stochastic bandit scenario. This intuition is confirmed by our results, in which the greedy and UCB strategy yield higher LP throughputs than any fixed pricer. On the contrary, for primal heuristics, it is sufficient to select successful ones more often, which is why a weighted sampling selection strategy can be preferred. In every case, we obtain considerable performance improvements on a diverse set of general MIP instances. While the work on the two heuristic frameworks is almost completed, the adaptive LP pricing is still prototypical. Future work on this requires to replace the measured solving time by a deterministic reward criterion. For the primal heuristics, it is interesting to compare the obtained results with new selection principles that are not based on past rewards.

Acknowledgements We thank Tobias Achterberg for useful comments and hints, especially with regard to Sect. 3.1. The work for this article has been partly conducted within the *Research Campus MODAL* funded by the German Federal Ministry of Education and Research (BMBF grant number 05M14ZAM).

References

1. Achterberg, T.: Conflict analysis in mixed integer programming. Discret. Optim. **4**(1), 4–20 (2007). https://dx.doi.org/10.1016/j.disopt.2006.10.006
2. Achterberg, T., Koch, T., Martin, A.: Branching rules revisited. Oper. Res. Lett. **33**(1), 42–54 (2005). https://dx.doi.org/10.1016/j.orl.2004.04.002
3. Berthold, T.: Heuristics of the branch-cut-and-price-framework SCIP. In: Kalcsics, J., Nickel, S. (eds.) Operations Research Proceedings 2007, pp. 31–36. Springer, Berlin (2008). https://dx.doi.org/10.1007/978-3-540-77903-2_5
4. Berthold, T.: Heuristic algorithms in global MINLP solvers, Ph.D. thesis, TU Berlin (2014)
5. Bubeck, S., Cesa-Bianchi, N.: Regret analysis of stochastic and nonstochastic multi-armed bandit problems. Found. Trends Mach. Learn. **5**(1), 1–122 (2012). https://dx.doi.org/10.1561/2200000024
6. Computational Optimization Research at Lehigh Laboratory (CORAL): MIP instances. https://coral.ise.lehigh.edu/data-sets/mixed-integer-instances/
7. Forrest, J.J., Goldfarb, D.: Steepest-edge simplex algorithms for linear programming. Math. Program. **57**(1), 341–374 (1992). https://dx.doi.org/10.1007/BF01581089
8. Gleixner, A., et al.: The SCIP Optimization Suite 5.0, Tech. Rep. 17–61, ZIB, Takustr. 7, 14195 Berlin (2017)

9. Harris, P.M.J.: Pivot selection methods of the devex lp code. Math. Program. **5**(1), 1–28 (1973). https://dx.doi.org/10.1007/BF01580108
10. MIPLIB – the Mixed Integer Programming LIBrary. miplib.zib.de
11. Witzig, J., Berthold, T., Heinz, S.: Experiments with conflict analysis in mixed integer programming. In: Salvagnin, D., Lombardi, M. (eds.) Integration of AI and OR Techniques in Constraint Programming, pp. 211–220. Springer, Cham (2017). https://dx.doi.org/10.1007/978-3-319-59776-8_17

Part XVII
Traffic, Mobility and Passenger Transportation

Integrated Optimisation for Descent Trajectory Operations and Airport Runway Assignment

Adrian Barea, Raul de Celis, and Luis Cadarso

1 Introduction

The current trends in air traffic development motivate the need for an update of the systems used for its management. Specifically, the air traffic growth poses a substantial challenge to the performance of air traffic flow management and control activities. Improvements in infrastructure and increases in manpower can help to mitigate this difficulty. However, this situation provides a significant opportunity to develop technologies in order to achieve levels of efficiency that comply with the future evolution of air transport industry by means of optimizing the operational performance and automatizing the processes of air traffic management.

Airport runway optimization is an ongoing challenge for air traffic controllers. There is a need to realize additional take-off and landing slots through better runway scheduling. Bennell et al. [2] review the techniques and tools of operational research and management science that are used for scheduling aircraft landings and take-offs. When one or more runways are in great demand, unnecessary delay and emissions may occur during peak periods while other runways at the same airport may be operating under capacity. The primary cause of this imbalance in runway utilization is that traffic flow is asymmetric (as a result of airline scheduling practices) and arrivals are typically assigned to the runway nearest the fix through which they enter the terminal area [4]. Delays and emissions are also incurred because arrival trajectories often include level segments and because precedence relationships. Kim et al. [4] present an optimization model for simultaneously assigning aircraft to

A. Barea · R. de Celis (✉) · L. Cadarso
European Institute for Aviation Training and Accreditation, Rey Juan Carlos University, Fuenlabrada, Madrid, Spain
e-mail: adrian.barea@urjc.es; raul.decelis@urjc.es; luis.cadarso@urjc.es

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_65

runways and scheduling the arrival and departure operations on these runways. Chandrasekar and Hwang [3] propose a framework to compute with computational efficiency, the optimal runway assignment, and sequencing of arrival and departure operations at an airport. Very closely to the runway assignment problem, the aircraft approach trajectory problem may be found, which is also heavily related to emissions and delays. Continuous descent arrival (CDA) procedures provide time and fuel savings, and reduce the noise impact of aircraft operations near airports [5]. The performance bounds of CDA procedure via multiphase optimal vertical trajectory generation problems with respect to two performance indices, flight time and fuel consumption, are investigated in [5].

This paper aims at integrating both problems: runway assignment and trajectory optimization problems. This integration aims at providing a novel scheme which eases the search of the global optimum, albeit it cannot be guaranteed due to inherent non-linearities, further reducing delays, emissions and noise in airport surroundings, as compared to a sequential solution approach. The model presented in this paper considers the final segments of approaching and landing, where airplane trajectories are contained in vertical planes, and as a result, aircraft behavior may be described with 3-D trajectories. Previous flight segments could be introduced by implementing 4-D trajectories linked to the ones already considered here. In parallel, runway assignment is performed by imposing time separation criteria and minimizing delays in order to optimize airport performance. Real data from "Adolfo Suarez Madrid-Barajas" airport is used so as to simulate commercial aircraft operations.

2 Mathematical Model

This section describes the mathematical model in detail. First, the objective function is described. Constraints follow.

Objective Function The purpose of this model is to minimize airplane delay and fuel consumption. As a result, objective function comprises a linear combination of airplane delay and negative values of aircraft mass at final time, which is directly related to fuel consumption.

$$min \left[\alpha \sum_{f \in F} r_f^+ - (1-\alpha) \sum_{f \in F} m_f (t_f = t_f^{final}) \right] \tag{1}$$

where F is the set of airplanes indexed by f and f_1, $\alpha \in [0, 1]$ is the weight that quantifies the relative importance of delay with respect to final mass in objective function, $r_f^+ \in [0, \infty]$ is the positive delay of airplane $f \in F$ with respect to its estimated landing time, m_f is the airplane mass and $t_f = t_f^{final}$ stands for the final time.

Kinematic Equations Equation (2) depict the vertical (h_f) and horizontal (x_f) displacements of the airplane as a function of speed (V_f) and descent angle (γ_f).

$$\left[\dot{x}_f, \dot{h}_f\right] = V_f \left[\cos(\gamma_f)\ , \sin(\gamma_f)\right]\ \forall f \in F \tag{2}$$

Forces and Aerodynamic Coefficients Modelling Lift (L_f), drag (D_f) and the polar (c_{df}) relationship between the drag and lift force (c_{lf}) coefficients are defined by (3) respectively.

$$\left[L_f, D_f, c_{df}\right] = \left[\tfrac{1}{2}\rho S_f (V_f)^2 \left(c_{lf}, c_{df}\right), c_{d_{0i_f}} + c_{d_{0_f}} + K_f (c_{lf})^2\right]\ \forall f \in F \tag{3}$$

where ρ is the air density, S_f is the wing surface, K_f is the induced resistance parameter and $c_{d_{0i_f}}$, $c_{d_{0_f}}$ are the parasitic drag coefficients of airplane.

Dynamics Equations Equation (4) determine the dynamics of an airplane whose trajectory is contained in a vertical plane, hypothesis that holds true for the final segments of approaching and landing, which are the ones considered in this model. They take into account airplane weight, thrust (T_f), aerodynamic forces, descent angle and the gravity acceleration (g).

$$\left[\dot{V}_f, \dot{\gamma}_f\right] = \left[\frac{T_f}{m_f} - \frac{D_f}{m_f} - g\sin(\gamma_f), \frac{L_f}{m_f V_f} - \frac{g}{V_f}\cos(\gamma_f)\right]\ \forall f \in F \tag{4}$$

Mass Variation Equations Equation (5) calculates the mass variation as a function of thrust force and specific fuel consumption parameters ($c_{f1_f}, m_{s2k_f}, c_{f2_f}$).

$$\dot{m}_f = -T_f c_{f1_f}\left(1 + V_f \frac{m_{s2k_f}}{c_{f2_f}}\right)\ \forall f \in F \tag{5}$$

Delay Equations Equation (6) computes the delay of airplane f. Such delay comprises the subtraction of two positive variables (r_f^+, r_f^-) so that only positive delays are considered in objective function. The reason why this behavior is desirable is that an airplane landing earlier than expected time (t_{e_f}) does not compensate for another one landing with delay.

$$r_f^+ - r_f^- = t_f^{final} - t_{e_f}\ \forall f \in F \tag{6}$$

Descent Profile Constraints Constraint (7) guarantees that airplane trajectories comply with height constraints imposed by the destination airport. Structure of $F(x_f)$ function is defined in later sections for Madrid Barajas airport.

$$h_f \geq F(x_f)\ \forall f \in F \tag{7}$$

Time Separation Constraints Equation (8) guarantee that airplanes landing in the same runway achieve a temporal separation (σ), typically 60 s, throughout their trajectory to ensure safety of operations, where δ_{f,f_1} is a binary variable that is 1 if two airplanes land in the same runway.

$$\left|t_{f_1}(x_{f_1} = x_f) - t_f(x_f = x_{f_1})\right| \geq \sigma\delta_{f,f_1}\ \forall f \in F,\ \forall f_1 \in F : f < f_1 \tag{8}$$

Speed Constraints Equation (9) states that speed at initial time ($t_f = t_f^{initial}$) is the corresponding airplane descent speed for Flight Level 100 and below (V_{des_f}) and that speed at final time is the corresponding aircraft approach speed in landing configuration (V_{app_f}).

$$V_f(t_f = t_f^{initial}) = V_{des_f}\ ,\ V_f(t_f = t_f^{final}) = V_{app_f}\ \forall f \in F \tag{9}$$

Runway Assignment Constraints Constraint (10) guarantees that each airplane is assigned to exactly one runway. N is the set of runways indexed by n and $\psi_{f,n}$ is a binary variable that is 1 if airplane f lands in the runway n.

$$\sum_{n \in N} \psi_{f,n} = 1\ \forall f \in F \tag{10}$$

3 Solution Approach: Time Discretization and Benders Decomposition

To formulate a Mixed Integer Nonlinear Programming model, differential equations (2), (4) and (5) are discretized: a set of time nodes is defined in which the aircraft position is defined by such equations. An adaptive-step (i.e., the integration step Δt is not a fixed parameter) implicit trapezoidal rule is chosen as integration scheme, which means the optimization model tunes the integration step as a function of the behavior of the rest of variables. In addition to the time discretization, it is assumed that function $F(x_f)$ of minimum height constraints (7) is a piecewise-defined continuous function. As a result, the problem is decomposed using the sets I, P and J. Problems $i \in I$ represent the different intervals in which function $F(x_f)$ of descent profile constraints (7) is defined. Phases $p \in P$ are subdivisions of $i \in I$ with different integration step Δt. Nodes $j \in J$ are subdivisions of $p \in P$ that constitute the time nodes for numerical integration of differential equations.

Benders decomposition [1] is a technique to solve problems with a structure such that, when some variables are fixed, the resulting problem is significantly easier to solve than the original one. Here, it allows to split the mixed integer nonlinear programming model into a master model, i.e., the runway assignment problem, and a sub-model, i.e., the aircraft descent trajectory problem. Both problems are solved iteratively, each one providing feedback to the other one, until a global or local

optimum is found. Although Benders decomposition is an exact method, it cannot guarantee global optimality due to non-linearities.

4 Computational Experiments

Real operational data of Madrid Barajas airport have been used for computational experiments. $F(x_f)$ in constraints (7), which defines minimum fly level, is a piecewise-defined function determined by a series of way-points (x_{wp}, h_{wp}) corresponding to the initial and final points of each of the pieces of the function. Figure 1 specifies way-point coordinates for Madrid-Barajas airport. Figure 2 depicts the number of aircraft to land at the airport as a function of estimated landing time. The model is coded in GAMS 25.1.1 and solved with CONOPT 4.05 and CPLEX 12.8.0 in a PC featuring an Intel Core i5-7500 and 8 GB RAM.

So as to obtain a solution of the complete test case, a rolling horizon approach is proposed. In this process, operations are partitioned in 30 min intervals and sequentially optimized while fixing the solutions of previous intervals. Moreover,

Parameter/Runway	18L(n=1)	18R(n=2)	32L(n=1)	32R(n=2)
$h^1_{wp}(m)$	2590.8	2895.6	1524	1524
$h^2_{wp}(m)$	1828.8	2133.6	1219.2	1524
$h^3_{wp}(m)$	1219.2	1524	-	-
$h^4_{wp}(m)$	585.83	606.86	589.17	574.85
$x^1_{wp}(m)$	740.8	0	0	2678.8
$x^2_{wp}(m)$	13334.4	12778.8	17680	11753.6
$x^3_{wp}(m)$	28891.2	23520.4	-	-
$x^4_{wp}(m)$	40373.6	40373.6	29162.4	29162.4

Fig. 1 Madrid-Barajas descent profile parameters

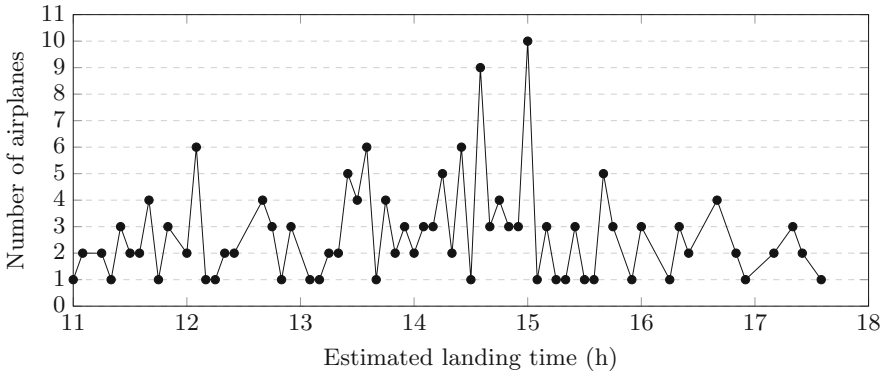


Fig. 2 Madrid-Barajas operations study case

two different operational scenarios have been considered. Scenario 1 represents a low-conflict situation where good-quality solutions are expected to achieve null total delay, on the contrary, scenario 2 simulates a high-conflict situation in which delays are hard to avoid. Table 1 shows the results of rolling horizon technique for both scenarios of South configuration (runways 18L, 18R). This configuration has been chosen because it is more complex than North configuration (runways 32L, 32R) so it is more suitable to test the performance of rolling horizon technique. Also, delay and consumed fuel of each airplane for both scenarios of South configuration are depicted in Figs. 3 and 4.

Table 1 Rolling horizon solution

Configuration	Scenario	Delay (s)	Consumed fuel (kg)	CPU time (s)	Gap
South	1	0	17,739.15	4819.98	2.5×10^{-4}
South	2	2659.06	17,911.79	3866.92	1×10^{-3}

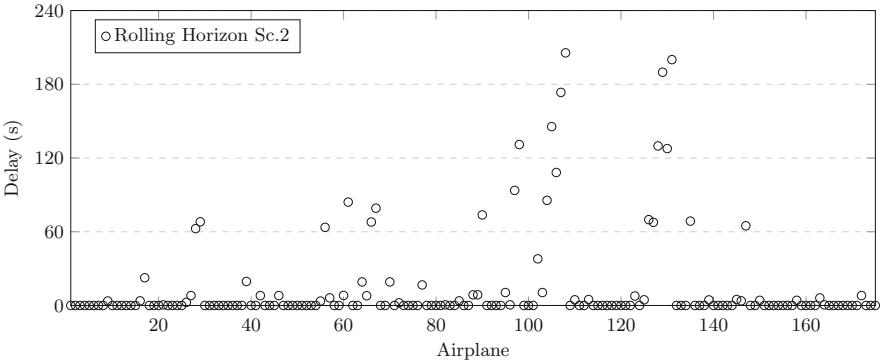


Fig. 3 South configuration delay

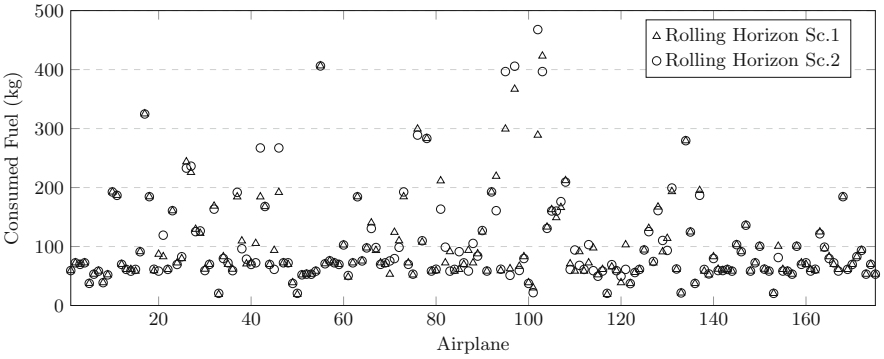


Fig. 4 South configuration consumed fuel

5 Conclusions

A mathematical formulation of a problem that unifies the computation of landing trajectories and runway assignment has been described. Such formulation has been discretized to obtain a mixed integer nonlinear programming problem, which is solved by means of a Benders decomposition. In order to test the performance of the proposed method, a large scale real operational case, corresponding to Madrid-Barajas airport is presented. To obtain a solution for the whole operational case, a rolling horizon technique, in which operations are partitioned in 30 min intervals and sequentially optimized, has been carried out, achieving accurate solutions in a practical computational time.

Acknowledgements This research was supported by the "Ministerio de Economía y Competitividad, Spain" (TRA2016-76914-C3-3-P) and "Consejería de Educación e Investigación, Comunidad de Madrid" (PEJD-2017-PRE/IND-4470).

References

1. Benders, J.F.: Partitioning procedures for solving mixed-variables programming problems. Numer. Math. **4**(1), 238–252 (1962)
2. Bennell, J.A., Mesgarpour, M., Potts, C.N.: Airport runway scheduling. Ann. Oper. Res. **204**(1), 249–270 (2013)
3. Chandrasekar, S., Hwang, I.: Algorithm for optimal arrival and departure sequencing and runway assignment. J. Guid. Control. Dyn. **38**(4), 601–613 (2014)
4. Kim, B., Li, L., Clarke, J.P.: Runway assignments that minimize terminal airspace and airport surface emissions. J. Guid. Control. Dyn. **37**(3), 789–798 (2014)
5. Park, S.G., Clarke, J.P.: Optimal control based vertical trajectory determination for continuous descent arrival procedures. J. Aircr. **52**(5), 1469–1480 (2015)

Real-Time Planning for Smart Charging of Electric Vehicle Fleets

Oliver Frendo and Nadine Gaertner

1 Introduction

1.1 Motivation

Numbers of electric vehicles (EVs) are increasing. Their growing need for electricity poses a challenge for both local infrastructures and the energy grid as a whole. Undersized connection lines and a lack of charging stations limit the number of EVs which can be charged concurrently. Combined with peaking power demands there is a need for smart charging. In company fleets, EVs accumulate for long stays in one location and smart charging can utilize this planning flexibility.

Currently, EVs start charging with full power as soon as they are connected to the charging station and are assigned on a first-come-first-serve basis. Most EVs in a typical company with 9–5 office jobs arrive at the same time, leading to peaks in demand at certain times. The charging infrastructure and its connection to the grid must be capable of handling said peaks. Additionally, the infrastructure's capacity is not being fully used at non-peak times, creating opportunity costs caused by not using unused capacity (i.e., to charge energy storage).

1.2 Smart Charging

Smart charging of electric fleets is a decision making problem regarding the charging processes of multiple EVs in the context of limited charging infrastructures.

O. Frendo (✉) · N. Gaertner
SAP SE, Walldorf, Germany
e-mail: oliver.frendo@sap.com; nadine.gaertner@sap.com

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_66

Limitations arise from insufficient numbers of charging stations and also from electrical installations with predetermined maximum loads. Flexibilities arise since the length of stay for the individual EV (i.e., a complete workday) typically exceeds the net charging time especially in the case of hybrid EVs. The degrees of freedom in the decision making problem therefore consist of the timing and power of the individual charging processes.

Smart charging involves both business and technical constraints. From an electrical engineering perspective, the loads from charging must be kept within the limits of a tree-structured installation of fuses. Charging involves three-phase electric power where imbalanced loads must be avoided. From an EV technology perspective, car model specific properties must be considered such as minimum and maximum charging currents, which phases can be used for charging as well as charging process suspendability and delayability. From an EV driver perspective, the main constraint is reaching a minimum state of charge (SoC) to guarantee trips before the next charging opportunity. Smart charging considers these constraints while creating a schedule which specifies which EVs may charge, at which charging station, when and at which current.

2 Model

We model the smart charging problem as a mixed integer linear program (MIP). The 24 h planning horizon is discretized into 96 time intervals of 15 min. Decision variables are the binary assignment variables $X_{i,n}$ of each EV n to each charging station i (similar to [2]) and charging power $P_{i,j,k}$ per charging station i, phase j and timeslot k. The following parameter subset shows the central constraints of the model.

$$a_{j,n} \in [0, 1] \quad \text{Ratio charged on phase } j \text{ for car } n \quad (1)$$

$$b_n \in \mathbb{R}+, \quad \text{Total charging need per car } n \text{ (unit: } Ah) \quad (2)$$

$$d_{k,n} \in \{0, 1\} \quad \text{Car } n \text{ availability for timeslot } k \quad (3)$$

$$h_{l,j} \in \mathbb{R}+ \quad \text{Fuse limit on level } l \text{ on phase } j \text{ (unit: } A) \quad (4)$$

The model contains a set of constraints for car-to-charging station assignments (5), (6). Knapsack-type constraints (7) reflect load limitations due to the fuse tree. Note that the $PhaseMap$ function reflects an electrical installation peculiarity where the individual phases in the charging station are connected to different phases of the overall grid. The big-M-method is applied to model the applicability of charging restrictions only for charging stations in use where the charging power is non-zero. This leads to complex formulations of upper and lower bound constraints for minimum and maximum charging powers and also for car specific

constraints such as satisfaction of charging needs, power ratios between phases (8) and suspendable charging.

$$\sum_i X_{i,n} \leq 1 \qquad \forall n \qquad \text{Each car } n \text{ max 1 station (5)}$$

$$\sum_n X_{i,n} * d_{k,n} \leq 1 \qquad \forall i,k \qquad \text{Each station } i \text{ max 1 car during } k \text{ (6)}$$

$$\sum_i P_{i,PhaseMap(i,j),k} \leq h_{l,j} \qquad \forall l,j,k \qquad \text{Fuse tree (7)}$$

$$a_{2,n} * P_{i,1,k} - a_{1,n} * P_{i,2,k} + M * d_{k,n} * X_{i,n} \leq M \qquad \forall i,k,n \qquad \text{Ratio 1st/2nd phase UP bound (8a)}$$

$$a_{2,n} * P_{i,1,k} - a_{1,n} * P_{i,2,k} - M * d_{k,n} * X_{i,n} \geq -M \qquad \forall i,k,n \qquad \text{Ratio 1st/2nd phase LO bound (8b)}$$

We consider a planning horizon of one day. In contrast to previous work [2–4] we consider phase wise power assignment. In the context of three-phase electrical power we schedule power individually per phase instead of summing all phases.

For the objective function, different components can be considered. (a) Electricity consumption costs apply time-dependent electricity prices linearly to consumed quantities and enforce charging at low-cost times. (b) Fair share describes scheduling each EV to reach a minimum SoC. The distance of the planned SoC to the minimum SoC is considered as costs. (c) Peak charges are determined by the highest load peak and their minimization ensures an even energy consumption over time. (d) Lastly, load imbalance costs describe differences between power assignments per phase.

In this work we optimize a weighted combination (9) of electricity costs z_1 and fair share z_2 with weights w_1, w_2 while satisfying fuse and car model limitations.

$$min : w_1 * z_1 + w_2 * z_2 \quad \text{with} \quad z_1 = \sum_i \sum_j \sum_k 0.25 * c_k * P_{i,j,k} \quad (9)$$

3 Method

In practice, the MIP model cannot be solved exactly in real time. Therefore we propose a two-step approach as illustrated in Fig. 1. First, during day-ahead planning we create an overall charging plan which is persisted to a database. We use another two-stage approach where an assignment heuristic fixes most integer decision variables ($X_{i,n}$) and a submodel of the MIP is solved. Second, our real-time planning heuristic makes use of the pre-computed schedule and uses it as guiding

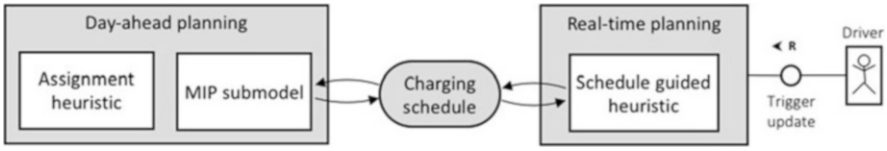


Fig. 1 Overall approach: a pre-computed charging schedule is updated continuously

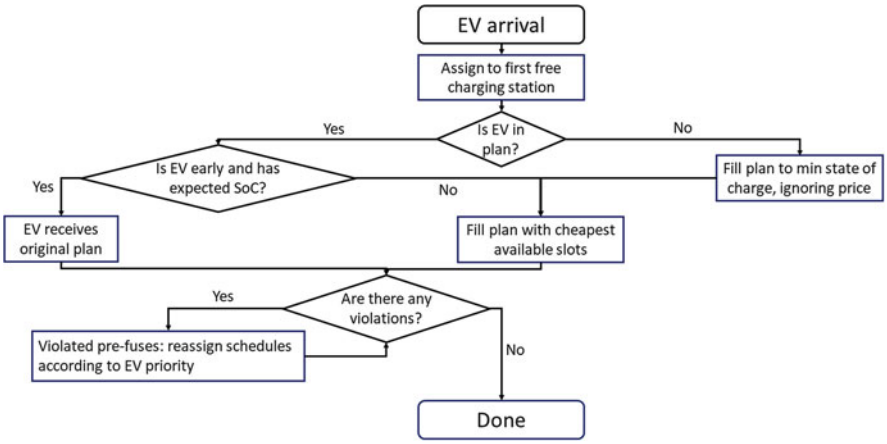


Fig. 2 Schedule guided heuristic: assigns EVs and adapts charging schedules

information. We refer to it as the schedule guided heuristic. An important feature of the heuristic is fast computation time for real-time decision making.

In this work we evaluate the real-time planning heuristic and compare it with the first-come-first-served approach which models the status quo by scheduling every EV for charging directly after arrival. If the overall capacity of the charging infrastructure is exceeded with the arrival of a new EV that EV does not charge.

The schedule guided heuristic proceeds as shown in Fig. 2. If an arriving EV is present in the day-ahead schedule it is assigned as planned. If it is an unexpected EV it is assigned to a charging station on a first-come-first-serve basis. The individual charging schedule for this EV is determined by priority-based rules within the schedule guided heuristic. Preference is given to EVs with a high deficit in SoC relative to remaining time (similar to [1]).

4 Results

4.1 Experimental Setup

A simulation is used to compare real-time planning smart charging approaches and to empirically measure runtimes. The schedule guided heuristic is applied with and without a schedule from day-ahead planning to investigate the guiding effect of the pre-computed schedule. Without the schedule, every EV is considered an unexpected arrival.

Historical data from a company fleet with several hundred EVs and a well developed charging infrastructure in Germany is used to generate a charging infrastructure and EVs for simulation. Each simulation comprises a complete day. For energy prices, a sample day from historical data of the intraday energy market is used.

4.2 Simulation Outcomes

Experimental results in Table 1 show both improved fair share and reduced electricity costs for the schedule guided heuristic compared to first-come-first-served. The minimum SoC is the required SoC for the driver to drive home and is set to 0.5 of the maximum capacity.

Both improvements of the schedule guided heuristic stem from charging schedules being spread out over the course of a day, thus allowing additional EVs to charge and selecting charging slots with lower electricity prices.

The schedule guided heuristic with day-ahead planning further improves on this by controlling EV assignment to charging station, while the other two approaches assign EVs on a first-come-first-served basis.

Table 1 Qualitative results (10 cars, 5 charging stations, average over 20 runs)

Method	Power (MWh)	Cost (€/MWh)	Min SoC reached	Final SoC reached
First-come-first-served	0.250	58.49	0.9145	0.8742
Schedule guided heuristic	0.269	52.24	0.9886	0.9179
Schedule guided heuristic + day-ahead schedule	0.286	49.90	0.9886	0.9543

For reference: a model of this size (used for day-ahead planning) consists of 13.176 continuous variables, 1.530 binary variables and 103.077 restrictions

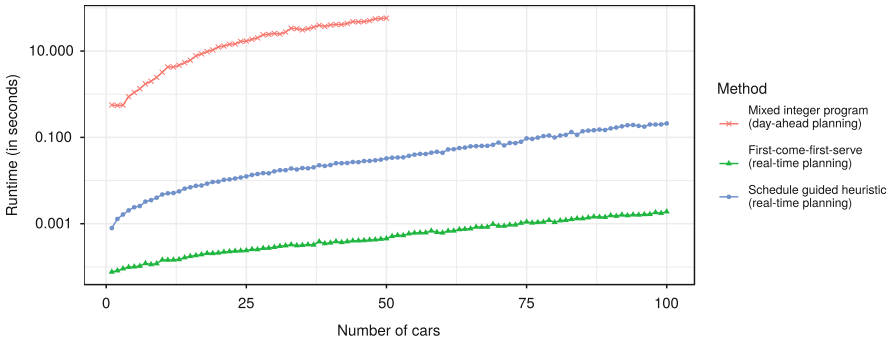


Fig. 3 Runtimes for increasing number of cars. The number of charging stations is held at 5. Each setting is an average of 20 runs. The three approaches compared above are (a) MIP with assignment heuristic (day-ahead planning), (b) first-come-first-serve (real-time planning) and (c) the schedule guided heuristic (real-time planning)

4.3 Runtime

To measure runtime, SCIP version 4.0.1 was used on a server with 128 GB RAM and 2x Intel Xeon E5-2660 v4 2.00 GHz. Figure 3 shows how the runtime for each approach scales with the number of cars. As shown in the left graph, runtimes for the MIP model become prohibitively large for small problem sizes with less than 100 EVs (successful solving of the MIP with more EVs was unreliable with the given hardware). In contrast, the real-time approaches have lower computation times by orders of magnitude.

5 Conclusion

To summarize, we present a two-step approach to address the EV fleet charging problem. The first step consists of day-ahead planning. Resulting charging schedules are then used as input for the second step consisting of a schedule guided heuristic. Experimental results show significant runtime, fair share and cost improvements. In contrast to previous work, this paper considers the three phases of the energy grid separately and so enables load balancing.

Future work should focus on exploring further methods to increase the size and complexity of instances which can be handled in real-time.

References

1. Akhavan-Rezai, E., Shaaban, M.F., El-Saadany, E.F., Karray, F.: Priority-based charging coordination of plug-in electric vehicles in smart parking lots. In: ISGT 2014, pp. 1–5 (2014). https://dx.doi.org/10.1109/ISGT.2014.6816501
2. Clemente, M., Fanti, M.P., Ukovich, W.: Smart management of electric vehicles charging operations: the vehicle-to-charging station assignment problem. IFAC Proc. Vol. **47**(3), 918–923 (2014). https://dx.doi.org/10.3182/20140824-6-ZA-1003.01061
3. Detzler, S.: Lademanagement für Elektrofahrzeuge, Ph.D. thesis, KIT, Karlsruhe (2016)
4. Xu, Z., Hu, Z., Song, Y., Luo, Z., Zhan, K., Wu, J.: Coordinated charging strategy for PEVs charging stations. In: 2012 IEEE Power and Energy Society General Meeting, pp. 1–8 (2012). https://dx.doi.org/10.1109/PESGM.2012.6345045

A Data-Driven Optimization Approach to Improve Railway Punctuality

Florian Hauck and Natalia Kliewer

1 Introduction

Delayed passenger trains are still a major issue for railway companies and one of the main reasons for displeased customers. In 2017, only about 80% of all long-distance trains in Germany arrived on time. In previous years, the punctuality rate was similarly low, even though Deutsche Bahn only includes trains with a delay of at least 6 min into this statistic [1]. In some other countries as well, railway companies struggle to achieve high punctuality rates [2]. Therefore, the on-time performance of trains needs to be improved and our aim is to develop a data-driven approach to increase punctuality.

The general railway planning process is very complex and consists of different optimization problems. Lusby et al. divide the planning process into three levels with seven different phases that are subsequently solved. The strategic level deals with long-term decisions and involves resource acquisition. The focus of the tactical level lies in resource allocation, whereas the operational level deals with sudden daily incidents [3]. Many research projects have developed approaches to optimize the different phases of this process. Our approach enhances the existing process by adding a data-driven add-on to the timetable generation phase. Most current timetable generation approaches focus on allocating train trips in order to find robust solutions that satisfy the demand without violating capacity restrictions. Solution approaches for the train timetabling problem are discussed by Cacchiani et al. [4], Caprara et al. [5], Peeters and Kroon [6] or Nachtigall and Opitz [7], for example. The aim of this paper is to further improve the results of those approaches by considering historical delay information. Therefore, we start with a given feasible

F. Hauck (✉) · N. Kliewer
Freie Universität Berlin, Department of Information Systems, Berlin, Germany
e-mail: f.hauck@fu-berlin.de; florian.hauck@fu-berlin.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_67

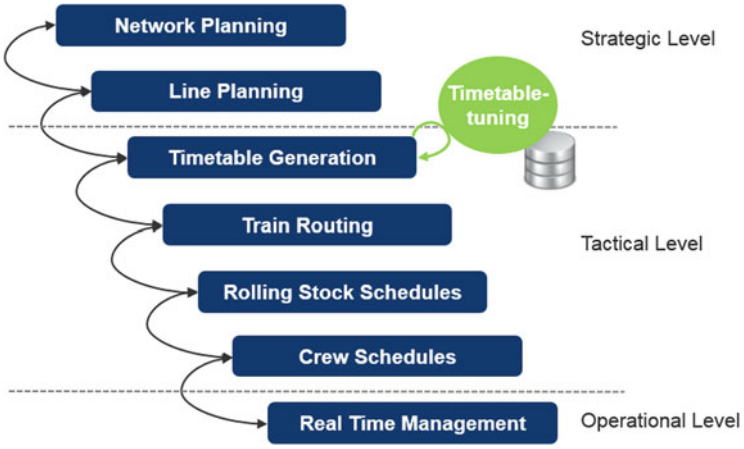


Fig. 1 The railway planning process (own illustration according to Lusby et al. [3])

timetable and subsequently improve it so as to avoid systematic delays that have appeared in recent years. This process is called timetable-tuning. In order to do this, we analyze train delay data from recent years and identify systematic delays. Next, small timetable adaptions are performed in order to avoid such systematic delays. In doing so, the timetable is converged to the real situation of past years. The planning process of Lusby et al. combined with our extension is shown in Fig. 1.

2 Optimization Model

The general idea of the presented approach is to calculate the average duration time of every train on every edge based on historic train data. Hereinafter, this duration is called the actual duration. Then, we compare the actual duration with the expected duration according to the current timetable. If the actual duration is longer than the expected duration, the train was often late in the past and might also be late in future trips. If the actual duration is shorter, the train is too early. Ideally, the two durations should be the same. Therefore, we minimize the positive gap between the actual and the expected duration. In doing so, all trips and all stations on the trips remain unchanged. We only adjust departure and arrival times and relocate buffer times. The loss of connections is allowed but every lost connection is penalized with individual penalty costs. Therefore, the original timetable is mostly preserved and tradeoffs between losing some connections for getting a better punctuality rate can be analyzed.

In order to apply our approach, we need the current timetable and historical trip data for at least one year. All other needed information can be derived from this input data. The definitions of sets, parameters and decision variables are given in Table 1. We assume that the timetable is periodic and use edges to model all train trips. An edge connects two nodes and every edge has a start event time and an end event time.

Table 1 Decision variables and parameters

Set	Description	Domain
T	Set of trips	
E	Set of edges	
WE	Set of waiting edges	$WE \subset E$
DE	Set of driving edges	$DE \subset E$
ET^t	Set of edges for trip t (ordered)	$\forall\, t \in T, ET^t \subset E$
WET^t	Set of waiting edges for trip t	$\forall\, t \in T, WET^t \subset E$
DET^t	Set of driving edges for trip t	$\forall\, t \in T, DET^t \subset E$
TE^e	Set of trips on edge e	$\forall\, e \in E, TE^e \subset T$
C^{te}	Set of connecting trips for trip t on waiting edge e	$\forall\, t \in T, \forall\, e \in WET^t, C^{te} \subset T$
Parameter	Description	Domain
$minD_{te}$	Minimum duration of trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
eS_{te}	Expected start time of trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
eE_{te}	Expected end time of trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
eD_{te}	Expected duration for trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
aD_{te}	Actual duration for trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
ω_{te}	Weighting factor for the actual duration of trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
$minC_{t_1 e\, t_2}$	Minimum connection time for trip t_1 on edge e to trip t_2	$\forall\, t_1 \in T, \forall\, e \in WET^{t_1}, \forall\, t_2 \in C^{t_1 e}$
$\rho_{t_1 e\, t_2}$	Penalty costs for missing the connection from t_1 to t_2 on edge e	$\forall\, t_1 \in T, \forall\, e \in WET^{t_1}, \forall\, t_2 \in C^{t_1 e}$
Variable	Description	Domain
xS_{te}	Change of the start time of trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
xE_{te}	Change of the end time of trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
ΔD_{te}	Gap between expected and actual duration of trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
yD_{te}	Indicator whether the actual duration is longer than the expected duration for trip t on edge e	$\forall\, t \in T, \forall\, e \in ET^t$
$yU1_{t_1 e\, t_2}$	Indicator whether start of trip t_1 on edge e is later than end of trip t_2 on edge e	$\forall\, e \in DE, \forall\, t_1 \in TE^e, \forall\, t_2 \in TE^e, (t_1 \neq t_2)$
$yU2_{t_1 e\, t_2}$	Indicator whether end of trip t_1 on edge e is sooner than start of trip t_2 on edge e	$\forall\, e \in DE, \forall\, t_1 \in TE^e, \forall\, t_2 \in TE^e, (t_1 \neq t_2)$
$yC_{t_1 e\, t_2}$	Indicator whether the connection from trip t_1 to trip t_2 on edge e is lost	$\forall\, t_1 \in T, \forall\, e \in WET^{t_1}, \forall\, t_2 \in C^{t_1 e}$

A node can represent stations or measuring points between stations. We use waiting edges to model the waiting time of a train inside a station while passengers exit or board the train. The decision variables represent positive or negative adjustments of starting or ending times. The model assumes that an edge can only be used by one train at a time.

The MIP is shown in Eqs. (1)–(13). The objective function is shown in Eq. (1). We minimize the positive gap between all actual and expected durations multiplied with a weighting factor ω. The weighting factor represents the accuracy of the actual duration times. Since the actual duration times are calculated using the average duration from historical data, some actual durations can have a high deviation and are therefore less useful. We only want to adapt the timetable to systematic delays; thus, actual durations with a low deviation are weighted higher. Additional costs are added to the objective value for every connection lost in the new timetable. Constraints 2, 3 and 4 make sure that the gap between the actual duration and the expected duration ($\Delta \mathrm{D_{t,e}}$) is 0 if the actual duration is smaller than the expected duration and otherwise the variable represents the value of the positive gap. This is necessary because we only want to minimize positive gaps. Negative gaps are not always bad because they can be used as buffer times. Constraint 5 effectuates that the duration of a train on a specific edge is at least as high as the minimum duration for this train on this edge. Constraint 6 guarantees that during each trip, the ending time of one edge equals the starting time of the following edge. Constraints 7, 8, 9 and 10 assure that every driving edge is used by maximal one train at once. For waiting edges, we allow an unlimited number of trains. Constraints 12 and 13 set the indicator variable $\mathrm{yC_{t_1,e,t_2}}$ to 1 if a connection is lost and to 0 if the connection remains.

$$\textbf{min} \sum_{t \in \mathcal{T}} \sum_{e \in \mathcal{ET}^{\mathrm{t}}} \Delta D_{t,e} * \omega_{t,e} + \sum_{e_2 \in \mathcal{WET}^{\mathrm{t}}} \sum_{t_2 \in \mathcal{C}^{\mathrm{e}_2}} \rho_{t,e_2,t_2} * yC_{t,e_2,t_2} \tag{1}$$

subject to

$$(aD_{t,e} - (eE_{t,e} + xE_{t,e} - eS_{t,e} - xS_{t,e})) * yD_{t,e} = \Delta D_{t,e} \quad \forall t \in T, e \in ET^t \tag{2}$$

$$aD_{t,e} - (eE_{t,e} + xE_{t,e} - eS_{t,e} - xS_{t,e}) \leq M * yD_{t,e} \quad \forall t \in T, e \in ET^t \tag{3}$$

$$(aD_{t,e} - (eE_{t,e} + xE_{t,e} - eS_{t,e} - xS_{t,e})) * yD_{t,e} \geq 0 \quad \forall t \in T, e \in ET^t \tag{4}$$

$$(eE_{t,e} + xE_{t,e}) - (eS_{t,e} + xS_{t,e}) \geq minD_{t,e} \quad \forall t \in T, e \in ET^t \tag{5}$$

$$eE_{t,e} + xE_{t,e} = eS_{t,e+1} + xS_{t,e+1} \quad \forall t \in T, e \in ET^t(e : 1..n-1) \tag{6}$$

$$((eS_{t_1,e} + xS_{t_1,e}) - (eE_{t_2,e} + xE_{t_2,e})) * yU1_{t_1,e,t_2} \geq 0 \quad \forall e \in DE, t_1 \in TE^e, t_2 \in TE^e(t_1 \neq t2) \tag{7}$$

$$((eS_{t_1,e} + xS_{t_1,e}) - (eE_{t_2,e} + xE_{t_2,e})) \leq M * yU1_{t_1,e,t_2} \quad \forall e \in DE, t_1 \in TE^e, t_2 \in TE^e (t_1 \neq t_2) \tag{8}$$

$$(eE_{t_1,e} + xE_{t_1,e}) - (eS_{t_2,e} + xS_{t_2,e}) * yU2_{t_1,e,t_2} \leq 0 \quad \forall e \in DE, t_1 \in TE^e, t_2 \in TE^e (t_1 \neq t_2) \tag{9}$$

$$(eE_{t_1,e} + xE_{t_1,e}) - (sS_{t_2,e} + xS_{t_2,e}) \geq -M * yU2_{t_1,e,t_2} \quad \forall e \in DE, t_1 \in TE^e, t_2 \in TE^e (t_1 \neq t_2) \tag{10}$$

$$yU1_{t_1,e,t_2} + yU2_{t_1,e,t_2} \geq 1 \quad \forall e \in DE, t_1 \in TE^e, t_2 \in TE^e (t_1 \neq t_2) \tag{11}$$

$$((eE_{t_2,e} + xE_{t_2,e}) - (eS_{t_1,e} + xS_{t_1,e}) - minC_{t_1,e,t_2}) * (1 - yC_{t_1,e,t_2}) \geq 0 \quad \forall t_1 \in T, t_2 \in C^{t_1^e}, e \in WET^{t_1} \tag{12}$$

$$((eE_{t_2,e} + xE_{t_2,e}) - (eS_{t_1,e} + xS_{t_1,e}) - minC_{t_1,e,t_2}) * yC_{t_1,e,t_2} \leq 0 \quad \forall t_1 \in T, t_2 \in C^{t_1^e}, e \in WET^{t_1} \tag{13}$$

3 Illustrative Example

We have linearized and implemented the presented optimization model using CPLEX and tested it with a small example in order to demonstrate its effectiveness. The example includes 4 trips, 14 edges and 8 connections and the planning horizon is divided into 13 time units. The illustration is shown in Fig. 2 on the top. For example, trip 1 starts at node 2 and ends at node 5 and has a waiting edge at node 4, where passengers can board or exit the train and change to trip 2 or trip 4. For each edge, the gap between expected and actual duration time is given (for simplicity, all actual durations are equally weighted here). The first starting times for trip 3 and trip 4 are fixed, but all other times are variable. The bottom graphic shows the optimal solution (with no penalty costs for lost connections) where all duration gaps have been removed and the new timetable has been adjusted to the historical delay data. However, now the connection of trip 4 to trip 1 is lost (because the connecting time here is at least two time units) as well as the connection of trip 1 to trip 3. With different penalty costs for lost connections different results can be achieved. It would also be possible to keep all connections but then a duration gap of two time units would remain. Through different penalty costs, tradeoffs between minimizing duration gaps and keeping connections can be analyzed.

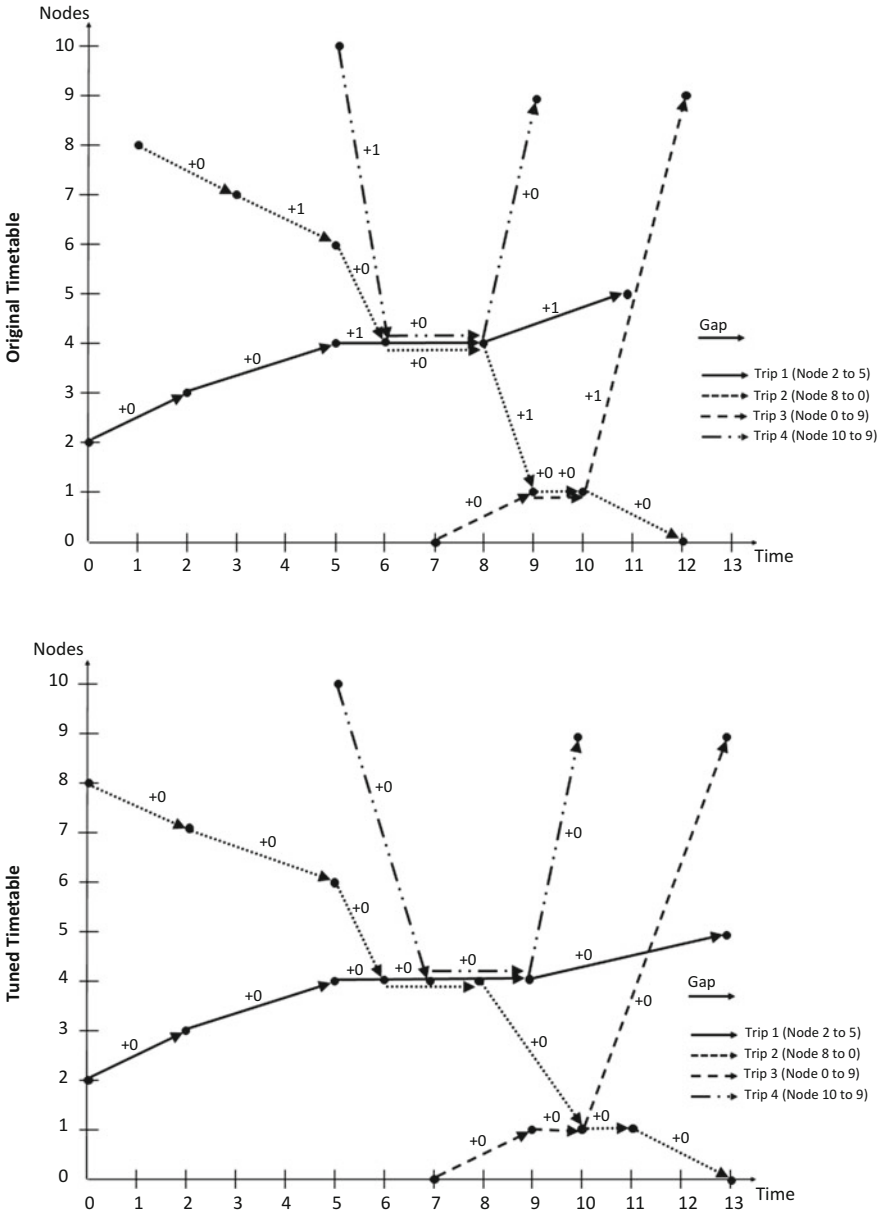


Fig. 2 Illustrative example

4 Conclusion and Outlook

The presented as the text size is less than 4 points. approach demonstrates how historical delay data can be used to improve railway punctuality. This approach can be integrated into the complex railway planning process and be used to further improve existing timetabling methods. The resulting schedule of other timetabling methods is used as an input for the presented approach and is then slightly adapted to match the average duration times as measured in recent years. Thereby, the benefits of timetabling methods are combined with the advantages of a data-driven delay analyzation approach. The presented example shows how the approach adapts a timetable to avoid systematic delays. In the future, we want to apply the approach to real data from Deutsche Bahn. Therefore, we use a dataset with delay information of 2016 and analyze the effects of this approach in a real-world example.

References

1. DB Group: Integrated report (2017). https://www1.deutschebahn.com/resource/blob/1639240/4bcaf98f54cf3ba4c84c83d4e0aea926/ib2017_dbkonzern_en-data.pdf. Retrieved 15 Jul 2018
2. EU flash barometer (2013). http://ec.europa.eu/commfrontoffice/publicopinion/flash/fl_382a_en.pdf. Retrieved 15 Jul 2018
3. Lusby, R.M., Larsen, J., Ehrgott, M., Ryan, D.: Railway track allocation: models and methods. OR Spectr. **33**(4), 843–883 (2011)
4. Cacchiani, V., Caprara, A., Toth, P.: A column generation approach to train timetabling on a corridor. 4OR **6**(2), 125–142 (2008)
5. Caprara, A., Fischetti, M., Toth, P.: Modeling and solving the train timetabling problem. Oper. Res. **50**(5), 851–916 (2002)
6. Peeters, L., Kroon, L.: A cycle based optimization model for the cyclic railway timetabling problem. In: Computer-Aided Scheduling of Public Transport, pp. 275–296. Springer, Berlin (2001)
7. Nachtigall, K., Opitz, J.: Solving periodic timetable optimisation problems by modulo simplex calculations. In: 8th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'08) (2008)

A Solution Approach for Railway Crew Scheduling with Attendance Rates for Multiple Networks

Julia Heil

1 Introduction

Since the 1990s, with the advances of computational power, computer-aided crew scheduling has gained momentum in the railway industry. By applying state-of-the-art Operations Research techniques, operators generate cost-efficient schedules, which fulfill desired levels of robustness and employee satisfaction at the same time. In recent years, as the size of railway crew scheduling problems increases in order to meet operators' practical needs, researchers developed solution methods for large-scale optimization in this field [4].

In the German regional railway passenger transport, state transport authorities tender railway networks publicly. Network parameters, such as geography, lines, timetable, quality of rolling stock and others, are specified in the transportation contract. Based on this contract and further restrictions, e.g., labor tariffs, railway operators plan their operations and make an offer ("planning for tender"). After offer acceptance, the nominated operator operates the network for the contractual period with recurring operational planning on (half-)yearly basis. Due to this process, operators plan and operate each single network separately. Only in few cases, based on their knowledge and experience, planners create duties across multiple networks manually. Assuming potential for further cost reduction, our research project investigates the effect on personnel cost of scheduling crews for multiple networks collectively.

Typically a crew in regional railway passenger transport consists of two types of workers: a train driver who operates the train and a conductor who, for instance, controls tickets or secures departures. In Germany, the conductors' presence is not

J. Heil (✉)
TU Dresden, Dresden, Germany
e-mail: jheil@tu-dresden.de; julia.heil@mailbox.tu-dresden.de

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_68

required for 100% of the train trips: State transport authorities define attendance rates per network, which vary depending on product type, time window and others. The attendance rate a is satisfied when the a-share of all kilometers assigned to attendance rate a are covered by at least one conductor and can be defined as $a \leq kilometers_{a,attended}/kilometers_{a,total}$.

Our work is based on the multi-period railway crew scheduling problem with attendance rates for conductors developed by Hoffmann et al. [3]. A hybrid column generation approach, which solves the pricing problem via a genetic algorithm, is applied. In this work, we present the extension to the multiple network problem and develop a two-phase optimization method to solve the corresponding very large-scale railway crew scheduling problem in reasonable time. We discuss computational experiments with a real case of 12 networks and determine the effect of multi-network crew scheduling on personnel cost.

2 Problem Definition

Crew scheduling is part of the operational planning process at a railway operator. Its objective is to generate a set of feasible duties, a *schedule*, which covers all trips at minimum cost. A *trip* symbolizes the atomic unit of a train run between two *relief points*. At a relief point, crew members can change vehicle, at some breaks are allowed. Hence, a *duty* is a combination of trips, which starts and ends at a *crew depot*. We define a single railway *network* as the entity of train lines with its corresponding parameters determined by the transportation contract. The *union of multiple networks* can be interpreted as a synthetic single network, which usually is very large and/or complex. Evidently, unifying single networks is only reasonable if they are interlinked, i.e. two networks share at least one relief point.

Both the entire schedule and an individual duty are limited by a number of various restrictions. This includes operational conditions, legal and work regulations, e.g., maximum duty time or break time rules, and contractual terms of the transportation contract, such as attendance rates. For simplification, we assume that the same restrictions apply to all networks with the exception of attendance rates, which must be satisfied for each network individually.

2.1 Mathematical Formulation

We formulate the multi-network crew scheduling problem with attendance rates as set covering model. Let M denote the set of trips i and N the set of duties j. For network-specific requirements, we define R as the set of networks r and G_r as the set of required attendance rates g of network r. Furthermore, let d_i be the travel distance of trip i and c_j the cost of the duty j. The binary assignment matrix A

defines the trips i covered by duty j. The model formulation is:

$$\min \sum_{j \in N} c_j x_j \tag{1}$$

$$s.t. \sum_{i \in M_r} d_{ig} y_i \geq g_r \sum_{i \in M_r} d_{ig} \quad \forall g \in G_r, \forall r \in R \tag{2}$$

$$\sum_{j \in N} a_{ij} x_j \geq y_i \quad \forall i \in M \tag{3}$$

$$y_i \geq a_{ij} x_j \quad \forall i \in M, \forall j \in N \tag{4}$$

$$x_j, y_i \in \{0, 1\} \quad \forall i \in M, \forall j \in N. \tag{5}$$

The objective function (1) minimizes the total cost of all duties. Constraint (2) ensures that the minimal required total distance of attended trips is satisfied for each attendance rate g of network r. Constraint (3) links the trips of a duty to the attendance rate constraint: if a duty covers trip i, the corresponding distance can add to the attendance rate constraint (2). Finally, constraint (4) ensures that if a trip is covered by at least one of the duties in the optimal solution, the distance of the trip add to the attendance rate fulfillment. In other words, the inequality of (3) allows deadheading, i.e. crew traveling on a train without being assigned to a work task, but only if the trip is already covered by another duty (cf. constraint (4)).

3 Two Solution Approaches

Railway crew scheduling problems are known to be very large and complex [5]. This is mainly because of the immense number of trips and an explosion of their combinatorial possibilities. For instance, in our data set, the average trip length is around 18 min. Hence, a cost-efficient duty of approximately 8 h combines on average 10 to 20 trips, in extreme cases up to 40. As a result, different approaches to manage the problem size and to reduce the computational time were presented in recent years. These approaches can be categorized as network-size reduction by trip combination (e.g., [5]) or as network decomposition into smaller sub-problems (e.g., [1, 4]). Additionally, acceleration techniques for solution methods, especially for the popular column generation heuristic, have been investigated extensively [2].

3.1 A Hybrid Column Generation Approach with Genetic Algorithm

Our work is based on a solution method developed by Hoffmann et al. [3]. It consists of a hybrid column generation approach with genetic algorithm ($CGGA$).

Table 1 A hybrid column generation approach with genetic algorithm

Step 0	*Initialization and initial set of duties.* Read input data and generate initial set of columns by combining trips to efficient blocks
Step 1	*Computation of dual multipliers.* Solve linear relaxed master problem with the current set of columns using a commercial solver and retrieve dual multipliers for trips
Step 2	*Generation of columns.* Generate columns with negative reduced cost using a genetic algorithm. If new columns with negative reduced cost are generated, go to Step 3 otherwise go to Step 4
Step 3	*Deletion of columns.* If a column was not part of the optimal solution in Step 1 for a defined number of iterations, delete it and return to Step 1
Step 4	*Optimal integer solution.* Generate integer solution of master problem using a commercial solver with defined stopping criterion

Acceleration techniques such as efficient trip combinations in the initial set of duties or column deletion are applied. For the sake of brevity, we outline the algorithm in Table 1 and refer the reader to Hoffmann et al. [3] for more details.

3.2 A Two-Phase Optimization Method

In order to accelerate computational time while maintaining high quality solutions, we propose a novel two-phase optimization method ($2PH$) based on a partitioning-and-re-combining strategy (see Fig. 1).

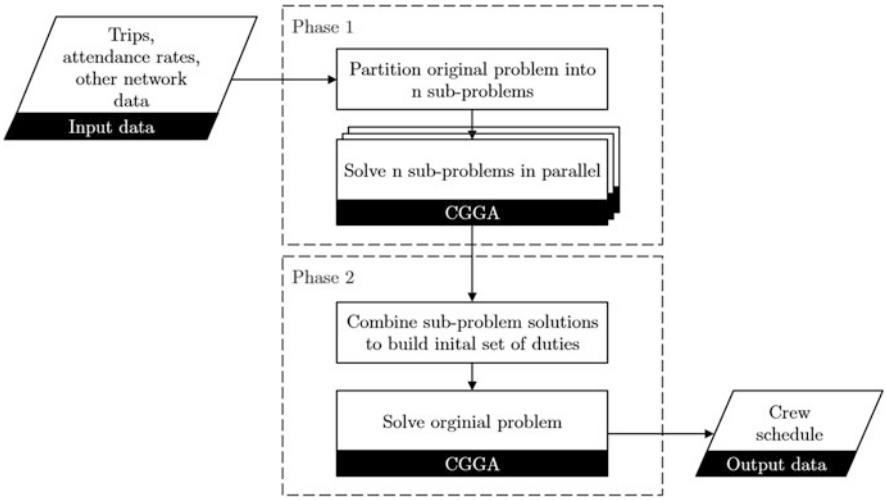


Fig. 1 Framework of the two-phase optimization method ($2PH$)

In the first phase (Phase 1), we decompose the original problem into smaller sub-problems, which are solved in parallel using $CGGA$. Subsequently, in Phase 2, we solve the original problem using $CGGA$ building on an initial set of duties, which combines the duties of each sub-problem solution and additional efficient trip combinations (cf. Step 0 of $CGGA$). Starting with the feasible and already good quality solution of Phase 1, $CGGA$ further improves the total schedule by creating duties across the sub-problems. The main advantage of this approach is that the generation of separate sub-problems enables parallel computation. The second phase ensures that the original problem is solved close to optimality and thereby all attendance rates requirements per network are fulfilled.

Sub-problems can be generated on the basis of different dimensions, such as network structure, geography and distance, time or historical schedule knowledge, e.g., the likelihood of two trips being combined in a duty [1]. Here, we take advantage of the underlying network structure and define a single network as a sub-problem.

4 Computational Experiments

We evaluate both approaches by means of a real data set of 12 interconnected networks. The 1-day problem consists of 6.491 trips and 136 relief points, thereof 28 are potential crew depots. The attendance rate distribution over all networks shows as follows: 79% of kilometers are assigned to an attendance rate of 100%, 3% to 50%, 13% to 20% and 6% to 0%. We manually add taxi trips (with attendance rate 0%) to ensure solution feasibility. All tests are executed on a Intel(R) Xenon(R) CPU E5-2630 with 2.6 GHz clock speed (384 GB RAM). The linear and integer programming problems are solved using the commercial solver Gurobi, version 7.5. We limit the computational time for the relaxed problem and genetic algorithm to 48 h. The integer programming model is terminated when the optimal solution is found (i.e. 0% gap between the best solution found and the current lower bound), at a gap $\leq$ 1% after 48 h or at the latest after 48 and 120 h in $2PH_{Phase1}$ and in $CGGA$ and $2PH_{Phase2}$, respectively.

Table 2 shows the average (*avg.*) as well as the minimal (*min.*) and maximal (*max.*) best objective value found (*Obj.*) of 10 test runs per solution approach. We also include the gap of the best integer solution found to the current lower bound (*Gap*), the total computational time (*Time*) and the share of mixed duties (*Mixed duties*), i.e. duties that consists of trips from two or more networks. In addition to the multi-network case, we solve each of the 12 single networks separately using $CGGA$. The objective values of those optimization runs add up to the best solution found of the single network scheduling case, which serves as reference point to evaluate the cost effect of multi-network scheduling (Δ_{Obj}).

As can be seen from Table 2, $2PH$ achieves a better objective value than $CGGA$ within less computational time (-53.72 h on average). A cause for this is the difference in problem size, i.e. the number of decision variables, between

Table 2 Computational results

	Solution approach		Obj.	Gap [%]	Time [h]	Mixed duties [%]	Δ_{Obj} [%]
Single networks	CGGA		7.775.820	–	–	–	–
Multiple networks	CGGA	Avg.	8.418.766	8.4	146.06	49.7	8.3
		Min.	8.313.055	8.1	167.98	46.5	6.9
		Max.	8.520.605	9.4	142.01	51.6	9.6
	2PH	Avg.	7.605.110	0.4	92.34	11.5	−2.2
		Min.	7.593.765	0.7	95.05	10.9	−2.3
		Max.	7.629.340	0.6	96.28	9.6	−1.9

$CGGA$ and $2PH_{Phase2}$, which both solve the original problem. $2PH_{Phase2}$ builds on a highly efficient and smaller initial set of duties. Therefore, a smaller number of decision variables can be maintained during the solution procedure. Hence, the number of iterations as well as the computational time per iteration is reduced.

Interestingly, the solution generated by $CGGA$ contains a high share of mixed duties (49.7% on average). However, considering the best objective value found and the remaining gap, this doesn't lead to a better objective value than $2PH$. Instead, with only 11.5% mixed duties, $2PH$ reduces the schedule cost by on average 2.2% in comparison to the single networks case.

5 Conclusion and Further Research

We presented a multi-network crew scheduling problem with attendance rates and compared two different approaches to solving the optimization problem. The two-phase optimization method based on a partitioning-and-re-combining strategy showed a better performance in both solution quality and computational time. We also showed that with multi-network crew scheduling personnel cost savings around 2% can be achieved.

Further studies aim to investigate the potential of different types of partitioning algorithms and metrics to further improve the performance of $2PH$. In this context, we address solving the crew scheduling problem of 12 networks for 1 week, the standard time period for crew scheduling of our partner company.

References

1. Abbink, E.J.W., van't Wout, J., Huismann, D.: Solving large scale crew scheduling problems by using iterative partitioning. In: Liebchen, C., Ahuja, R.K., Mesa, J.A. (eds.) 7th Workshop on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems (ATMOS'07), pp. 96–106. Leibniz Center for Informatics, Wadern (2007). https://dx.doi.org/10.4230/OASIcs.ATMOS.2007.1168

2. Desaulniers, G., Desrosiers, J., Solomon, M.M.: Accelerating strategies in column generation methods for vehicle routing and crew scheduling problems. In: Ribeiro, C.C., Hansen, P. (eds.) Essays and Surveys in Metaheuristics, pp. 309–324. Springer, Boston (2002). https://dx.doi.org/10.1007/978-1-4615-1507-4_14
3. Hoffmann, K., Buscher, U., Neufeld, J.S., Tamke, F.: Solving practical railway crew scheduling problems with attendance rates. Bus. Inf. Syst. Eng. **59**, 147–159 (2017). https://dx.doi.org/10.1007/s12599-017-0470-8
4. Jütte, S., Thonemann, U.: Divide-and-price: a decomposition algorithm for solving large-scale railway crew scheduling problems. Eur. J. Oper. Res. **219**, 214–223 (2012). https://dx.doi.org/10.1016/j.ejor.2011.12.038
5. Kwan, R.S.K., Kwan, A.S.K.: Effective search space control for large and/or complex driver scheduling problems. Ann. Oper. Res. **155**, 417–435 (2007). https://dx.doi.org/10.1007/s10479-007-0203-3

User-Based Redistribution in Free-Floating Bike Sharing Systems

Christoph Heitz, Roman Etschmann, Raoul Stoeckle, Thomas Bachmann, and Matthias Templ

1 Introduction

Free-floating bike sharing systems (BSS) are relatively new. In such systems, bikes can be dropped off at any location within a specified area. Bikes can be located and unlocked via smartphone. In recent years, free-floating BSS have been rapidly introduced worldwide, for example by *oBike* (China) or *LimeBike* (USA), and it seems as if the future of BSS are free-floating systems. In Zurich, the first free-floating e-bike system has been launched in 2017 by the insurance company La Mobilière and is now being operated by the company *smide* (www.smide.ch).

Free-floating systems avoid expensive docking stations and offer more flexibility. On the other hand, the redistribution of bikes from areas where they are dropped off to areas where they are needed is much more costly than for station-based systems, because the bikes are distributed over the complete area rather than concentrated at few stations. This is a particular critical issue since redistribution is typically the largest cost factor of a BSS even for station-based BSS (see, e.g [1]). In order to reduce the redistribution costs, several authors have suggested to transfer at least a part of the redistribution task to the users, which is called user-based redistribution. Nearly all of the literature, however, is focused on station-based systems (e.g. [2–6]), and to our knowledge nothing has been published on user-based redistribution for free-floating systems.

C. Heitz (✉) · R. Etschmann · M. Templ
School of Engineering, Zurich University of Applied Sciences, Winterthur, Switzerland
e-mail: heit@zhaw.ch

R. Stoeckle
smide KmG, Uster, Switzerland

T. Bachmann
La Mobilière, Bern, Switzerland

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_69

In this paper, we study the problem of user-based redistribution for free-floating BSS. We study the dynamics of flee-floating systems, and we assess the potential of dynamical user incentivation for generating user-based redistribution. The presented results have been derived during the development of the *smide* system, where user-based redistribution is a core element. *Smide* has been developed in a cooperation of the *Zurich University of Applied Sciences* and *La Mobilière*. The incentive approach has been tested in a field test in fall 2017, and is implemented in the current *smide* system.

2 Free-Floating BSS: Dynamics and User Incentives

2.1 Basic Dynamical Model of Free-Floating BSS

For analyzing user incentivation in BSS, we study a quadratic area A of 3 km × 3 km, discretized in quadratic cells of width c. We chose c = 100 m, but a finer grid can be used. Locations are specified by indices (i,j). The number of bikes is denoted by N. Demand for bikes generally depends on both time and location. For this paper, we restrict ourselves to constant and homogenous demand, thus in each cell we have the same demand rate.

If a demand occurs in cell (i,j), the user is willing to walk a certain distance to pick up an available bike. Empirical results show that the willingness to walk for picking up a bike is about 300–500 m [4]. We use a simple behavioral model assuming that all users are willing to walk a maximum distance of 2 cells in each direction. Thus, a user in cell (i,j) is willing to pick up a bike within the region U(i,j) consisting of all cells (i',j') with $i' \in [i-2, i+2]$, and $j' \in [j-2, j+2]$. The user picks the next available bike within U. A service violation occurs if U is empty. After the ride, the bike is dropped off in a random cell.

The state of the system is given by the bike distribution n(i,j,t). It is given by the following process: Demand occurs as a Poisson process in a random cell. If a bike in the environment is available, it is removed. The destination cell is chosen randomly and the bike is dropped off immediately. Incentives might change the drop-off location. Similar to [4, 5], we define the service level β as the probability that a demand is met, i.e. that a bike is available within the walking distance.

The studied dynamical system is balanced: pick-up rates and drop-off rates are equal in each cell. This is unrealistic, of course, but we chose this configuration because it allows to study the effect of user-based redistribution directly, without the need of explicitly modeling the operator's redistribution activity. The modeled situation mimics a redistribution strategy where the operator focuses his redistribution activity on counterbalancing macroscopic and systematic flows, but ignores the stochastic fluctuations of the bike distribution.

2.2 *Service Level and System Dynamics of Free-Floating BSS Without Incentives*

As a reference case, we study a system without any user incentives by simulating the above specified process with 50,000 demand events. The trip time has been set to zero for simplicity. In Fig. 1 (left), the service level is shown as a function of the number of bikes N. For N = 100 bikes (11.1 bikes/km^2), the service level is 82%, and it approaches 97% for N = 400.

Note that the spatial distribution of bikes shows a strong spatial correlation of neighboring cells: The probability of a cell being empty is higher if the neighboring cells are empty, and vice versa. A result is the appearance of large starved areas (Fig. 1 right, top). This is caused by the following mechanism: Users within a starved area walk to the border of the area to pick up a bike. Thus, the demand for bikes just outside of a starved area is higher than on average, and a starved area tends to grow. In addition, if a bike is being dropped in the middle of a starved area, it collects all the demand of the neighboring cells, and is being picked up much faster than on average, making starved areas more persistent. Thus, the fact that users walk to pick up a bike creates a bike distribution that shows large empty areas combined with overpopulated areas elsewhere. In Fig. 1, right, bottom, the bike distribution is shown for the case where users would *not* be willing to walk. It can be seen that the distribution is much more regular, and there are much fewer areas where no bike can be found within U. The corresponding service level is much higher, (Fig. 1, left, orange dotted line). Thus, the pick-up walking of users has a significant impact

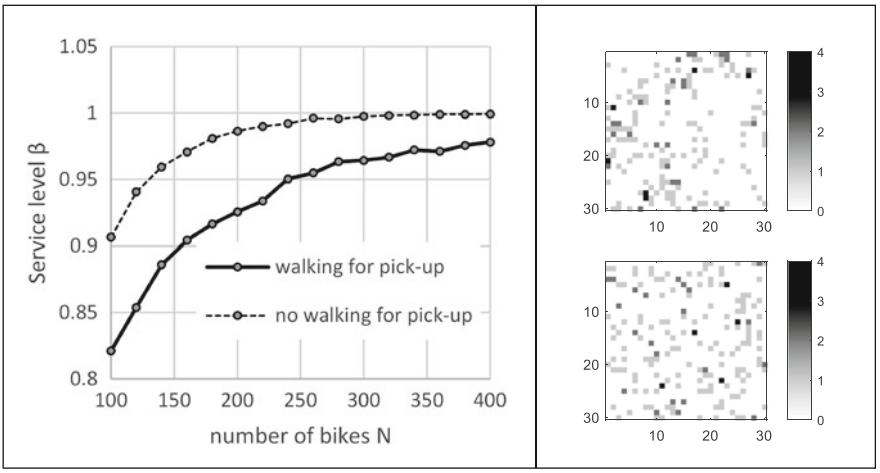


Fig. 1 Left: Service level as a function of number of bikes N with users walking to pick up a bike (solid line) vs. not walking (dotted line). Right: Sample bike distributions for N = 200 for walking users (top) vs. non-walking users (bottom)

on the service level by creating this spatial correlation. It is specific to free-floating BSS, and is not present in station-based BSS.

2.3 Incentive System: Value Function

User incentives for improving the bike distribution may be offered in different ways: Users may be motivated to drop off bikes in starved areas, which leads to an immediate service level improvement, or to pick up bikes in overpopulated areas, hoping that the bike increases the service level at the drop-off location, or a combination of both (giving incentives for a specific trip). In this paper and as in [5], we only study drop-off incentives, as they are the most direct way for improving service level.

The goal of such an incentive system is an improvement of the service level by influencing where users drop off their bikes after usage. A dropped bike at location (i,j) has a positive impact on the service level if and only if it avoids a service failure in the future. The earlier the bike is needed, the more value is generated by the drop-off in terms of service level. Thus, we base our incentive approach on the time interval Z(i,j) until a bike dropped at (i,j) will actually be needed, i.e. the time until a service failure would occur if the bike was *not* dropped.

Z(i,j) is a random variable and depends on the current bike distribution as well as the future demand and the dropping behavior. We define a *value function v(i,j)* as the reciprocal of the expectation value of Z(i,j): $v(i,j) = (E(Z(i,j)))^{-1}$. So, v(i,j) is a measure that directly reflects the impact of a drop at (i,j) on the service level. Maximizing this value for each ride would lead to a minimization of service failures and, equivalently, to an improvement of the service level.

The quantity E(Z(i,j)) is well defined. It could be determined by forward-simulating the system starting from the current bike distribution until the next service failure occurs due to a demand event at (i',j'), where (i',j') ϵ U(i,j). By averaging many simulation runs, the expectation value E(Z(i,j)) could be determined. This, however, is numerically cumbersome. In the following we derive a method of calculating an approximation of E(Z(i,j)).

Let B(i',j') the time to the next service failure created by a demand event in cell (i',j') ϵ U(i,j). The time to the next service failure that could be avoided by dropping a bike at (i,j) is the minimum of all these B(i',j'). If the B(i',j') were exponentially distributed random variables, then the minimum of a set of such variables would be the exponentially distributed, where the rate is a the sum of the rates of the B(i',j'). This motivates setting the value v(i,j) to

$$v\,(i,\,j) = \sum_{(i',j') \in U\,(i,\,j)} \frac{1}{E\,(B\,(i',\,j'))} \tag{1}$$

For calculating E(B(i',j')), we approximate the dynamics of the full system by restricting the dynamical analysis to the environment U(i',j'). Let n(t) be the number of bikes in this area, with n(0) the current number of bikes, calculated by summing up the current bike distribution in U(i',j'). We assume that n(t) is a stochastic birth-death-process with a birth rate λ_U and death rate μ_U, where λ_U is the aggregation of the demand in U(i',j'), and $\mu_U = \beta \bullet \lambda_U$, where β is the service level. The expectation value of B(i',j') can then be calculated as a mean first passage time of an equivalent Markov model with absorbing state -1. In general, the transient probabilities can be calculated by integrating the Komogorov forward equations. For the special case of constant rates, the expectation value can be calculated analytically [7].

$$E\left(B\left(i', j'\right)\right) = \frac{n(0) - 1}{\lambda_U - \mu_U} = \frac{n(0) - 1}{(1 - \beta)\,\lambda_U}. \tag{2}$$

Inserting this in Eq. (1), the value v(i,j) of dropping off a bike at a specific location (i,j) can be calculated for each location.

2.4 *Incentive System: Rewards for Triggering Behavioral Changes*

The goal of each incentive system is to change the behavior of the users for improving the service level. In this paper, our aim is not to analyze a specific incentive system, but to assess the potential of an incentive system based on the value function v(i,j) as defined in Sect. 2.3, in a most general way. We assume that the value map v(i,j) is mapped into a reward function r(i,j) where r is strictly monotonically increasing with v, and the reward function is communicated continuously to all users via their smart phone. This is in contrast to [4, 5], where drop-off rewards are user-specific and offered only after the true destination is known, which leads to complex implementation issues. Our approach offers a reward for each trip in advance, and the reward only depends on the dropping location. This leads to additional expenses, as rewards are also given if the user does not change her behavior. On the other hand, such a system may trigger additional rides to starved areas. Is shown in Sect. 2.5, this contributes substantially to the improvement of the service level.

There are still many degrees of freedom for concrete implementation, for example the type of the reward (e.g. monetary refund or free minutes for the next ride), its level, and how exactly the mapping between v and r is defined. For our analysis we only assume that users are offered more reward if v is higher, and that the reward is sufficiently high to influence the users' behavior. Based on empirical findings in the field test of *smide*, we assume that offering a location-based drop-off incentive has two different behavioral effects: (a) Users may drop off their bike somewhere close to, but not directly at, their destination, if they can increase their reward by doing so. So, they would accept to take a short walk at the end of the

trip for increasing their reward. (b) The offered rewards might trigger additional trips. For example, users may choose to use the BSS instead of public transport, just because they can earn a reward which can be used for future BSS trips.

We assume furthermore that a user either ignores the offered reward altogether, or he changes his behavior such that his reward is maximized. For case (a), a user with destination (i,j) would drop the bike at the cell (i',j') ϵ U(i,j) where v is maximum. For case (b), the user would make a trip with a destination corresponding to the global maximum of v. Note that, in contrast to [4, 5], we do not try to calculate explicitly the optimum level of the reward. The percentage of users responding to the incentive system is a model parameter which accounts for the dependence of behavioral change on the reward magnitude, averaged over all users. Optimizing the way of setting the height of the rewards with methods like in [4, 5] might reduce the costs of the incentive system, but would not substantially change the overall effect of the incentive, which is what we are focusing on in this paper.

Using these behavioral assumptions, we are able to generate quantitative results about the potential of an incentive system based on the value function v, without the need to specify exactly how the value map is transferred into corresponding rewards.

2.5 Simulation Results

We simulate the user behavior under the above described incentive scheme. Trips are generated as described in Sect. 2.1. With a probability p_1, the user is reacting to the incentive by dropping the bike in the environment U(i,j) around his destination (i,j), choosing the cell with the highest value. In addition, a proportion p_2 of additional trips are generated. For example, $p_2 = 0.1$ means that 10% additional trips are generated by offering the incentives, compared to not offering incentives at all. For these additional trips, we assume that the pick-up distribution is the same as before, but the drop-off location is the location with the highest reward within the whole area. This is motivated by our empirical findings with users of *smide*, indicating that additional trips are generated by high rewards only.

In Fig. 2, the service level for different parameter values are shown. It can be seen that both mechanism of behavioral change lead to substantial increase of the service level, and for achieving a given service goal, the number of needed bikes can be strongly reduced. For a service level goal of 0.95, without user incentives, around 250 bikes would be needed for the simulated area of 9 km^2. It can be seen that, even with a modest reaction of users to the incentives, a reduction of bikes by 50% can be obtained. For example, with 10% additional trips and a 30% user response for drop-off displacement of 300 m, the number of bikes can be reduced by 50%.

Of course this depends on the fraction of users that react to the incentives. Many parameters might influence this, such as level and type of incentive (e.g. free minutes vs. monetary reward), but also the service design and the user communication. In field tests of *smide* during fall 2017 in Zurich, we tested the response of the users

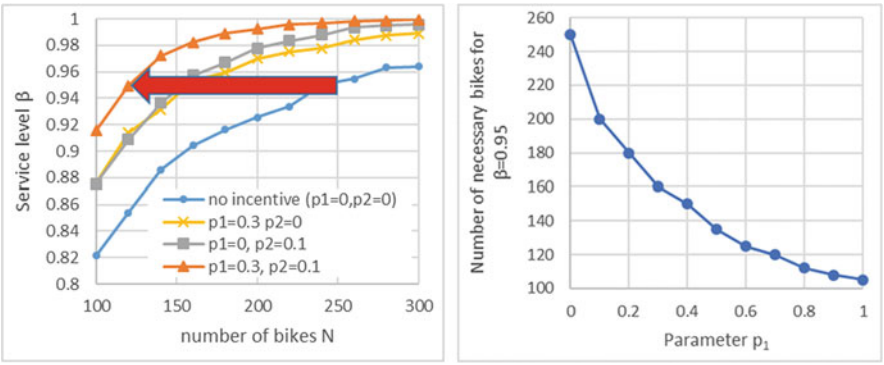


Fig. 2 Left: Service level as function of number of bikes for different parameter values of p_1 and p_2. For a service level goal of 95%, the number of necessary bikes can be reduced by 50% with even modest participation of users. Right: Sensitivity analysis for p_1 (with $p_2 = 0$)

to incentives consisting of 5 . . . 15 free minutes for future rides, offered at specified locations. This generated about 10% additional trips, so $p_2 = 0.1$ seems realistic.

No empirical results are yet available for parameter p_1. In Fig. 2, right side, a sensitivity analysis with respect to p_1 is shown, indicating the number of bikes necessary to achieve a service level of 95% as a function of p_1, where p_2 is set to 0. It can be seen that the effect of user participation is highly non-linear: Even a small part of the users can have a large impact on the system performance. In our case, a participation of 20% already yields 50% of the effect. This supports earlier findings (see, e.g [4]).

The simulation results indicate that both behavioral effects as described in Sect. 2.3 seem to have a similar efficiency, but their mechanisms differ: Effect (a) is efficient because it reduces the spatial correlation. For this, even small changes of the dropping location are sufficient. Effect (b), on the other hand, leads to a large-scale redistribution of bikes.

3 Discussion

Our results suggest that user-based redistribution has a huge potential in free-floating BSS. Only considering drop-off incentives, we have shown that even with moderate participation of users, the number of necessary bikes to reach a specified service level can be substantially reduced. Since the total costs of BSS, to a large extent, are proportional to the number of deployed bikes, this directly translates in huge savings. Of course, incentives also incur costs. However, in the *smide* field test it turned out that the savings were a factor 8 higher than the incentive costs.

In this paper, we only showed results for constant and homogeneous demand and dropping patterns. However, the same approach was applied to time-varying and non-homogeneous patterns with very similar results.

References

1. Guía metodológica para la implantación de sistemas de bicicletas públicas en España. Instituto para la Diversificación y Ahorro de la Energía (IDAE), Madrid (2007)
2. Reiss, S., Bogenberger, K.: Optimal bike fleet management by smart relocation methods. 2016 IEEE 19th Int. Conf. Intell. Transp. Syst. **1**, 2613–2618 (2016)
3. Todd, M., Xue, L., Barth, M.J.: User-based vehicle relocation techniques for multiple-station shared-use vehicle systems, TRB (2004)
4. Singla, A., Santoni, M., Bartók, G., Mukerji, P., Meenen, M., Krause, A.: Incentivizing users for balancing bike sharing systems. AAAI. **1**, 723–729 (2015)
5. Pfrommer, J., Warrington, J., Schildbach, G., Morari, M.: Dynamic vehicle redistribution and online price incentives in shared mobility systems. IEEE Trans. Intell. Transp. Syst. **15**(4), 1567–1578 (2014)
6. Fricker, C., Gast, N.: Incentives and redistribution in homogeneous bike-sharing systems with stations of finite capacity. EURO J. Transp. Logist. **1**, 1–31 (2014)
7. Jouini, O., Dallery, Y.: Moments of first passage times in general birth–death processes. Math. Methods Oper. Res. **68**(1), 49–76 (2008)

Data Analytics for Trajectory Selection and Preference-Model Extrapolation in the European Airspace

Carlo Lancia, Luigi De Giovanni, and Guglielmo Lulli

1 Introduction

Air Traffic Management (ATM) systems have to face the continuous growth of air transportation demand, leading to increasing congestion of the airspace. As stated by modern ATM concepts like Trajectory Based Operations (TBO) [7], the definition of daily flight trajectories, while guaranteeing safety, has to trade-off the need of individual airspace users to optimize their operations and the objective of reaching optimum performance of the whole ATM network. The problem is known as Air Traffic Flow Management (ATFM). In this context, mathematical models aiming at supporting its solution should take airspace users' preferences into account. The scope of this paper is the definition of a methodology to capture the information on airspace users' preferences and to embed it into mathematical models for ATFM, in particular the ones based on the selection of one trajectory for each flight chosen from a set that may be given a-priori, or dynamically determined by the optimization process (e.g. [2, 8]). In particular, given a flight and related airspace user, we want to determine a measure of the preference of the flight for each alternative trajectory.

Preferences depend on several factors such as trajectory geometry, speed, fuel consumption, en-route charges, weather conditions, business model and specific objectives (e.g., legacy air carriers may prefer short routes, whereas some low-

C. Lancia (✉)
Mathematical Institute Leiden University, Leiden, The Netherlands
e-mail: c.lancia@math.leidenuniv.nl

L. De Giovanni
Università degli studi di Padova, Padova, Italy
e-mail: luigi@math.unipd.it

G. Lulli
Lancaster University Management School, Bailrigg, Lancaster, UK
e-mail: g.lulli@lancaster.ac.uk

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_70

cost carriers may lean toward longer routes to avoid high en-route charges). Some determinants are only partially known or unknown, as they are part of confidential business information. As a consequence we propose a data driven approach to extract consolidated knowledge on airspace user preferences from historical information on flight trajectories. Recent literature proposes several works devoted to the analysis of historical flight trajectories and, among the ones that are more closely related to our research, we cite the following. In [4], hidden Markov model, clustering and regression are combined towards trajectory prediction and balanced use of airspace capacity. A statistical analysis of the relations among trajectory length, duration, fuel cost, en-route charge and other possible determinants in the European airspace is presented in [1]. In [6], clustering, linear regression and multinomial logit models are used to identify nominal trajectories per origin-destination pair and explain en-route inefficiency.

We focus on preference modeling and apply data analytics and machine learning tools with the objective of identifying the preference parameters that directly supports mathematical models for ATFM. In Sect. 2, we describe the proposed methodology, starting from data available from data repositories, and applying route clustering and classification to learn route choice determinants and related preference parameters. Section 3 briefly discusses two sample applications in the European airspace. Section 4 concludes the paper and outlines future research.

2 Data and Methods

We propose a method to learn preferences for flight trajectories from historical data. In particular, we refer to the European airspace and consider Eurocontrol DDR2 repository [3]. Among other information, this contains a full description of the trajectory filed, for each flight, at the pre-tactical stage (Filed Tactical Flight Model). Trajectory data are longitudinal and include, for each element in the sequence, latitude, longitude, flight level and the time at which that point has to be flown (4D trajectories). We consider all flights operated between a fixed origin-destination pair: to this end, both origin and destination are considered as a set of one or more airports serving the same area (for example, Rome would include both Fiumicino and Ciampino airports).

As a first step, we perform a clustering to determine groups of homogeneous 4D trajectories, based on their geometry and operating speed: since trajectories in the same cluster are similar to each other, we assume that they have the same preference levels. We apply a methodology similar to the one proposed in [5] and used in [6]. First, we resample each trajectory by linear interpolation to obtain a same-length description of all trajectories: each trajectory is described by a number of 4D-points equal to $2M$, where M is the number of points in the longest original trajectory. Each of the $4 \times 2M$ features is shifted to mean zero and scaled to unit variance, and 10% of points are trimmed off trajectory head and tail to exclude take-off and approach. Next, Principal Component Analysis (PCA) is used to reduce dimensionality by retaining the N first components that explain at least a pre-specified fraction of

the observed variance. The choice of the variance-thresholds sensibly affects the final results and we perform a calibration analysis on this parameter. Trajectory groups are obtained by density-based clustering (DBSCAN) in $\mathbb{R}^{4N}$. This method is appealing because it works well with the complex geometry of the data and can discriminate outliers.

The second step uses a tree classifier to learn how the features of a specific flight are related to cluster membership. We train the classifier on the following features: day of the week, week number (for seasonal effects), part of the day (night, early morning, late morning/early afternoon, late afternoon), airline code, airline type (legacy/low-cost), and aircraft model. We use one-hot encoding for categorical variables. We recall that a tree classifier (see Fig. 2) produces a binary tree with internal nodes representing a condition on a feature that can be true (left branch) or false (right branch), and leaves specifying a cluster: running across the tree from the root, each flight is classified according to the cluster associated to the reached leaf. The tree is validated by k-fold cross-validation and its precision and recall are cross-checked with a different classification approach based on a Support Vector Machine (SVM), trained on the same features (and validated by same k-fold cross-validation).

The third step uses the tree to classify all the flights in the dataset, counting, for each leaf l and each cluster c, the number of flights belonging to c and reaching l. By normalizing the flights count in each leaf, we obtain the required measure between 0 and 1 of the preference level of each cluster as a function of the features above. For example, with reference to Fig. 2, a flight reaching the third leaf (from top) has a preference of 0.69 for all trajectories that can be included into cluster 1, 0.22 for cluster 0, 0.09 for cluster 3, 0 for others.

In order to obtain some preliminary insights into the tree structure, we explore univariate associations between flight features using Cramer's V-index with the Bergsman's bias correction. This index is based on the Pearson's chi-squared statistic and measures the association between two categorical variables on a scale between 0 (independence) and 1 (maximum intercorrelation).

3 Results and Analysis

The proposed methodology has been applied to two sample case studies in the European airspace: the origin-destination pairs Rome-Paris and Istanbul-Frankfurt. The analysis is based on flights operated in the period from June 15 to September 15, 2016, with data extracted from Eurocontrol DDR2. Rome includes Rome Fiumicino (LIRF) and Rome Ciampino (LIRA) airports, Paris includes Paris Charles de Gaulle (LFPG), Paris Orly (LFPO) and Beauvais–Tillé (LFOB), Istanbul includes Istanbul Atatürk (LTBA), Frankfurt includes Frankfurt am Main (EDDF). The first scenario considers 1957 flights (1219 from LIRF to LFPG, 566 from LIRF to LFPO, 171 from LIRA to LFOB and 1 from LIRF to LFOB), the second scenario has 930 flights. All analysis are performed with Python 3.6.4, Scikit-learn 0.19.1, and Basemap 1.1.0.

Figure 1a and b show the clustering result for Rome-Paris and Istanbul-Frankfurt, respectively. Trajectory clusters are presented with their projection on a map and

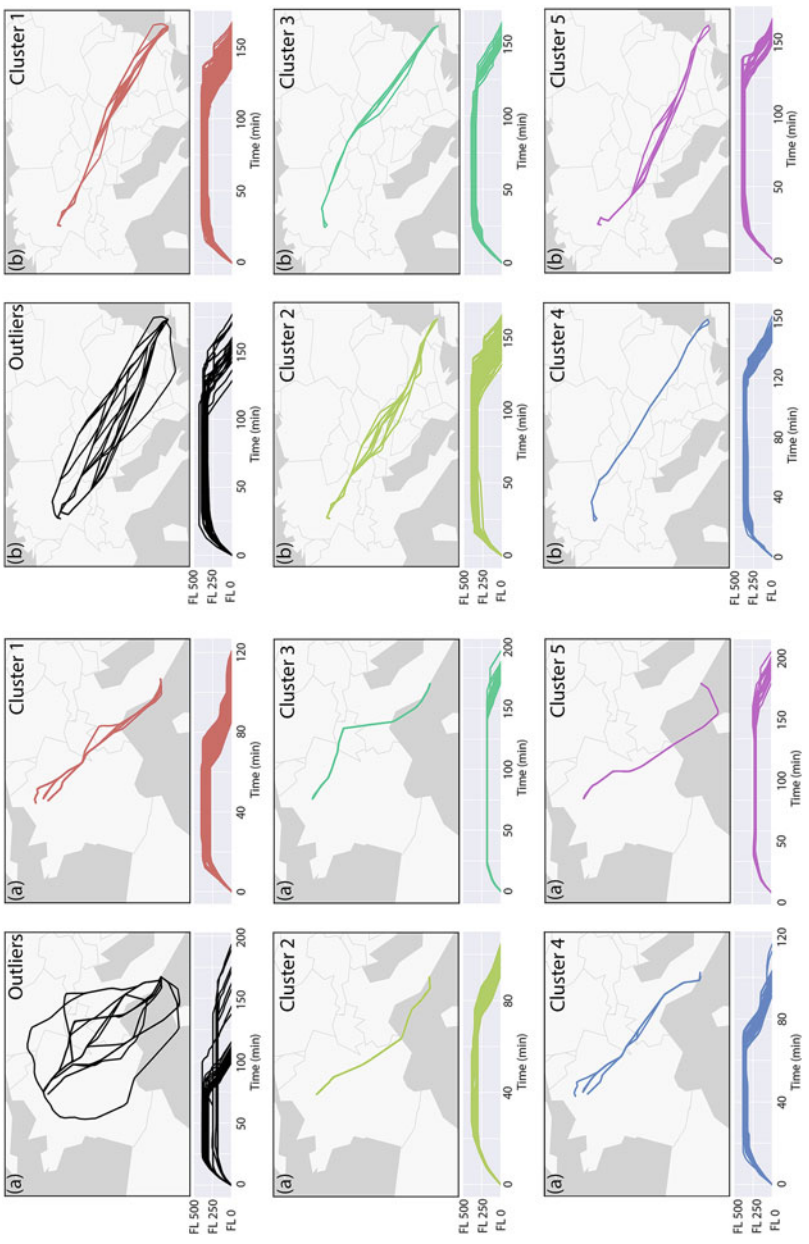


Fig. 1 Clustered trajectories for Rome-Paris (**a**); available online at https://ibb.co/iz46wo) and Istanbul-Frankfurt (**b**); available online at https://ibb.co/ncRx2T). Clusters are displayed as a 2D map projection and flight level profile over time

Table 1 Cluster composition

	Outliers	Cluster 0	Cluster 1	Cluster 2	Cluster 3	Cluster 4
Rome-Paris	31	1616	223	32	39	16
Istanbul-Frankfurt	22	519	310	37	25	22

the altimetric profile over time. Clusters generally look well-defined for Rome-Paris, while they are visually less separated for Istanbul-Frankfurt, which suggests more variation in the flight altitude/speed. Table 1 shows the count of flights in each cluster. The clustering result is affected by both the proportion of variance explained by the PCA and the hyperparameters of DBSCAN. For this analysis, we tried to achieve a clustering where clusters (1) have a larger size than the outliers group and (2) are visually homogeneous in the plane projection. The idea is that a lower variance-threshold, e.g. 0.85, yields a rougher representation of trajectories, so that selecting a lower maximum-distance ε in DBSCAN should give a good trade-off between cluster separation and size of the outliers.

In a 5-fold cross-validation, the performance of the tree classifier are more than satisfactory in both scenarios: for Rome-Paris, the mean values of precision and recall are 0.917 (standard deviation 0.006) and 0.941 (s.d. 0.003); for Istanbul-Paris, the values of precision and recall are 0.666 (s.d. 031) and 0.708 (s.d. 0.017). These results are in line with those obtained with a SVM trained on the same feature set. Figure 2 shows the decision tree for Rome-Paris (the figure is also available at ibb.co/hq3mGy). For the sake of space, the decision tree for Istanbul-Frankfurt is omitted here, and available on line at ibb.co/hUTWid. For Rome-Paris, the tree is rather simple: 3 internal levels and 8 leaves, with levels of preference that in each leaf are generally concentrated on a single cluster. This is expected from the reported levels of precision and recall, and is due to the presence of a very strong association, revealed by the Cramérs's V-index (Table 2), between cluster and airline type ($V = 0.62$) and code ($V = 0.57$). For Istanbul-Frankfurt, the tree is more complex: 5 internal levels and 17 leaves. Lower precision and recall translates into a less unbalanced preference distribution at each leaf. The main difference between this tree and the former is the stronger role of seasonality: many splits are performed based on week number and month. This can be appreciated also from the values of Cramér's V, which are lower than before for airline code and type, and of comparable magnitude for all features. As a note on Table 2, the association between aircraft model and cluster might be confounded by the relationship between aircraft model and airline code (the related index is $V = 0.75$ for Rome-Paris and $V = 0.84$ for Istanbul-Frankfurt).

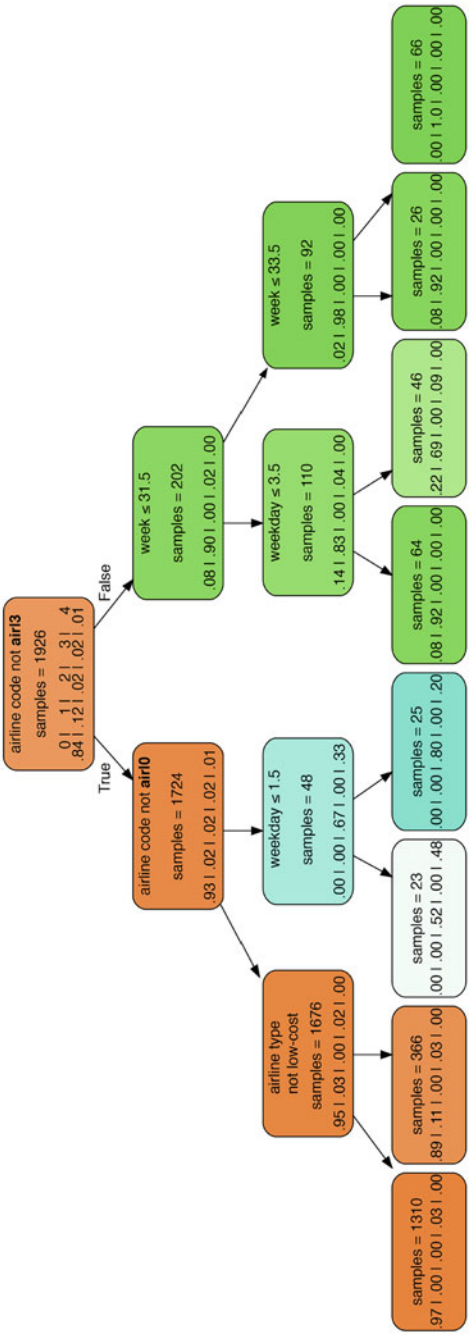


Fig. 2 Decision tree for Rome-Paris. Airline names are masked

Table 2 Cramér's V for the univariate associations between trajectory representative and model features in each scenario

	Airl. code	Airl. type	Aircraft model	Day part	Weekday	Week	month
Rome-Paris	0.57	0.62	0.45	0.18	0.07	0.05	0.01
Istanbul-Frankfurt	0.28	0.17	0.22	0.11	0.10	0.20	0.18

4 Conclusions

We presented an innovative approach to the definition of airline preferences based on machine learning. The idea is to learn homogeneous trajectories via clustering of historical flight data, and to explore the relation between preference and flight features with a decision tree. We illustrated the methodology in two different scenarios and cross-checked the results of the tree with a SVM. The appealing property of the decision tree is that the composition of each leaf can be directly interpreted in terms of preference for a flight to trajectories in each cluster.

Further developments of the method might include evaluating the use of ensemble tree classifier like *adaBoost* to overcome the limitations of the current approach in case a larger set of features is used. A multivariate generalized linear model might shed more light into the relationship between trajectory preference and flight features. This would represent a step towards a deeper analysis of trajectory determinants and would explain the rationale of flight preferences.

The proposed approach should be envisioned in the final goal of feeding ATFM models that include information about flight preference and measuring the impact of airline preferences to the solutions of realistic scenarios. In this respect, this work is a stepping stone for future research in this direction.

References

1. Delgado, L.: European route choice determinants. In: Eleventh USA/Europe Air Traffic Management Research and Development Seminar (2015)
2. Djeumou Fomeni, F., Lulli, G., Zografos, K.G.: An optimization model for assigning 4D-trajectories to flights under the TBO concept. In: Twelfth USA/Europe Air Traffic Management Research and Development Seminar (2017)
3. Eurocontrol: DDR2 Reference Manual (2014)
4. Fernández, E.C., Cordero, J.M., Vouros, G., Pelekis, N., Kravaris, T., Georgiou, H., Fuchs, G., Andrienko, N., Andrienko, G., Casado, E., Scarlatti, D.: DART: a machine-learning approach to trajectory prediction and demand-capacity balancing. In: SESAR Innovation Days, Belgrade, pp. 28–30, November 2017
5. Gariel, M., Srivastava, A.N., Feron, E.: Trajectory clustering and an application to airspace monitoring. IEEE Trans. Intell. Transp. Syst. **12**(4), 1511–1524 (2011)
6. Liu, Y., Hansen, M., Lovell, D.J., Chuang, C., Ball, M.O., Gulfing, J.M.: Causal analysis of en route flight inefficiency - the US experience. In: Twelfth USA/Europe Air Traffic Management Research and Development Seminar (2017)

7. SESAR: The roadmap for delivering high performing aviation for Europe. European ATM Master Plan. Executive view. Edition (2015)
8. Sherali, H.D., Smith, J.C., Trani, A.A.: An airspace planning model for selecting flightplans under workload, safety, and equity considerations. Transp. Sci. **36**(4), 378–397 (2002)

Periodic Timetabling with 'Track Choice'-PESP Based on Given Line Concepts and Mesoscopic Infrastructure

Raimond Wüst, Stephan Bütikofer, Severin Ess, Claudio Gomez, Albert Steiner, Marco Laumanns, and Jacint Szabo

1 Introduction

In the operational management of railway networks, an important requirement is the fast adaptation of timetable scenarios, in which operational disruptions or time windows with temporary unavailability of infrastructure, for instance during maintenance time windows, are taken into account. In those situations, easy and fast reconfiguration and recalculation of timetable data is of central importance. This local and temporal rescheduling results in shifted departure and arrival times and sometimes even in modified stop patterns at intermediate stations of train runs. In order to generate reliable timetabling results it is a prerequisite that train-track assignments, as well as operational and commercial dependencies are taken into consideration and that all these dependencies are not conflicting with each other. Hence, finding the right level of detail for modelling track infrastructure and train dynamics is crucial for supporting the planning process in an optimal way. This requirement motivated several research groups to combine common timetabling procedures with constraints resulting from mesoscopic infrastructure information in recent years.

R. Wüst (✉) · S. Bütikofer · S. Ess · C. Gomez · A. Steiner
Institute for Data Analysis and Process Design, Zurich University of Applied Sciences ZHAW, Winterthur, Switzerland
e-mail: wura@zhaw.ch

M. Laumanns
Bestmile SA, Lausanne, Switzerland
e-mail: marco.laumanns@bestmile.com

J. Szabo
IBM Research, now at Google Switzerland, Zurich, Switzerland
e-mail: szabojacint@gmail.com

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_71

From the existing approaches, we will discuss below some that are relevant to our work. Hansen and Pachl [1] show how running, dwell and headway times at critical route nodes and platform tracks must be taken into account for train processing and present a deep timetable quality analysis depending on these parameters. De Fabris et al. [2] calculate arrival and departure time, platform and the route in stations and junctions that trains visit along their lines. Bešinović et al. [3] present a micro–macro framework based on an integrated iterative approach for computing a microscopically conflict-free timetable that uses a macroscopic optimization model with a post-processing robustness evaluation. Caimi et al. [4] extend PESP (see e.g [5]) by proposing the flexible periodic event scheduling problem (FPESP), where intervals are generated instead of fixed event times. By applying FPESP, the output does not define a final timetable but an input for finding a feasible timetable on a microscopic level, ([4, 6]).

Our modelling approach is also based on an extension of PESP and takes the service intention (SI) as input data structure. The SI was first described in Caimi [6] and integrates commercial timetabling requirements given by the respective line concept on one side and technical constraints on the other. It largely corresponds to the 'line concept', and represents functional timetabling requirements including line data, line frequencies and separations as well as line transfers at specific stations. Similarly to de Fabris et al. [2], we call this level of abstraction of the available resources 'mesoscopic topology'. Together with the functional requirements of the SI this mesoscopic infrastructure data model of a given scenario is entered into a standard timetable editor (see, e.g. SMA Viriato, [7]).

2 Methodology

The investigation of feasible event times for individual train runs and the corresponding resource allocations fitting into an integrated clock face timetable is usually done manually in a time consuming way. On the other hand, algorithmic approaches for solving this task computationally require models based on microscopic information about track capacity. This capacity information can be aggregated to headway constraints that are used for solving standard periodic timetable problems. In order to facilitate this step, we present a generic approach, which makes use of the mesoscopic infrastructure. We call this approach Track-Choice PESP (TCPESP), as it can be considered as an extension of PESP, which includes the selection of relevant headway constraints into the optimization problem.

The SI is defined by a set of train runs. Each train run belongs to a line L and is characterized by the sequence of sections that are traversed and a corresponding time interval, which is required for either running or stopping on a corresponding track section. Each time interval has a minimal and maximal value. Stop nodes typically provide a service for boarding or de-boarding. Together, a pair of train runs moving in opposite directions makes up a train circulation.

In the TCPESP model, the mesoscopic infrastructure consisting of sections is summarized as a set I of operation points. Operation points are largely track sections and stations but can also be other critical resources as junctions (see example below). As mentioned before, each operation point $i \in I$ is associated to a capacity consisting of a set of tracks T_i. A train run $l \in L$ is described by a sequence of operation points of I.

Based on our mesoscopic model we form an event-activity network (E, A). The set E of events consists of an arrival event arr_{li} and a departure event dep_{li} for each train run $l \in L$ and operation point $i \in l$. The activities $a \in A$ are directed arcs from $E \times E$ and describe the dependencies between the events. For every train run we have arcs between arrival and departure events at the same operation points (dwell times or trip times) and arcs between departure and arrival events of successive operation points (time needed for the trip between operation points). Further arcs include connections between train runs, headways and turnaround operations (see Sect. 3). We refer to [5] for a detailed overview of the modelling options of dependencies. Figure 1 provides a sample of such an event graph.

Headway arcs $a \in A_H$ are especially important for explaining the 'track-choice PESP model' below. Headway arcs are used to model safety distances between trains running in the same and in opposite directions (see example in Fig. 1). For the sake of simplicity we consider in TCPESP (1) below only headways related to one operation point, i.e. we omit headways for train runs in opposite directions over several successive operation points. The problem formulation (1) can be easily extended to include general headways.

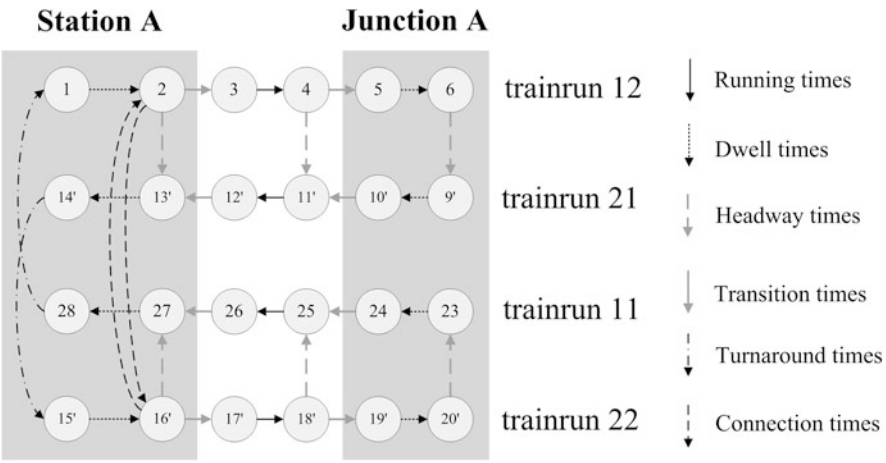


Fig. 1 Sample of an event activity network, where arcs connect arrival and departure events. Nodes belonging to grey shaded boxes indicate events at operation point type operation points. Other nodes indicate track type arrival and departure events. Arrow line styles indicate different types of time dependencies

The classical PESP tries to determine a periodic schedule on the macroscopic level (i.e. without using the tracks at an operation point) within a period T. Event $e \in E$ takes place at time $\pi_e \in [0, T)$. The schedule is periodic with time T, hence each event is repeated periodically $\{\ldots, \pi_e - T, \pi_e, \pi_e + T, \pi_e + 2T, \ldots\}$.

The choices of the event times π_e depend on each other. The dependencies are described by arcs $a = (e,f)$ in A and modeled as constraints in the PESP. The constraints always concern the two events e and f and define the minimum and maximum periodic time difference l_a and u_a between them. These bounds are given as parameters in the PESP model. We therefore look for the event times π_e for every $e \in E$ that fulfill all constraints of the form

$$l_a \leq \pi_e - \pi_f + p_a T \leq u_a, \text{ for all } a = (f, e) \in A, \tag{1}$$

where p_a is an integer variable that allows the constraints of the form (2) to be met in a periodic sense.

Track-Choice PESP Model We extended the classical PESP model by using the number of tracks T_i at each operation point $i \in I$. The track-choice PESP model assigns the arrival event arr_{li} and the departure event dep_{li} of train run l at operation point i uniquely to a track in T_i. We can use these assignments to switch on headway arcs $a \in A_H$ by using the following big-M-approach. In addition to variables π and p from the classical PESP model we need: (i) Binary variables tc_{et} (track choice) for each event $e \in E$ and track $t \in T_{i(e)}$, where operation point $i(e)$ is associated to event e, i.e. e is equal to arr_{li} or dep_{li} for a train run l. (ii) Binary variables h_a for every headway edge $a = (f, e) \in A_H$. Headway edges are always between events at the same operation point, therefore $T_{i(e)} = T_{i(f)}$ holds. The track-choice model is defined by:

$$\min \; f(\pi, p)$$

$$s.t. \;\; l_a \leq \pi_e - \pi_f + p_a T \leq u_a, \quad \forall \; a = (f, e) \in A \setminus A_H, \tag{2}$$

$$l_a - (1 - h_a) M \leq \pi_e - \pi_f + p_a T \leq u_a + (1 - h_a) M, \quad \forall \; a = (f, e) \in A_H, \tag{3}$$

$$\sum\nolimits_{t \in T_{i(e)}} tc_{et} = 1, \quad \forall \; e \in E, \tag{4}$$

$$tc_{arr_{li}t} = tc_{dep_{li}t}, \quad \forall \; l \in L, i \in l, t \in T_i, \tag{5}$$

$$h_a \geq tc_{et} + tc_{ft} - 1, \quad \forall \; a = (f, e) \in A_H, t \in T_{i(e)} \tag{6}$$

$$tc_{et}, h_a \in \{0, 1\}, \pi_e \in [0, T), p_a \in \mathbb{Z}, \quad \forall \; e \in E, t \in T_{i(e)}, a \in A,$$

where M is a big enough natural number.

There are many different objective functions $f(\pi, p)$ described in literature [7]. In our test case below we minimize the total passenger travel time. In (2) are the normal PESP constraints summarized (without headway arcs). In (3) are the headway constraints, which can be switched off with a big-M technique. The assignment of the events to the tracks is done in (4). (5) is used to assign the corresponding arrival and departure events to the same track. In (6) the headway variable is set to 1, if the events take place on the same track, i.e. the headway is required at this operation point.

3 Case Study

In order to validate the proposed TCPESP model we designed a simple test case. The relationship between the macroscopic timetable events of three train lines are illustrated by means of a simplified network graph (see Fig. 2a). To validate the model, a virtual railway network was defined for which the service intention was implemented (see Fig. 2b).

The test network contains the two main station nodes (Station A and Station B) connected by line 2, and three stop stations, Stop A, Stop AT (served by line 1) and Stop BT (served by line 3). The planning-relevant secondary conditions for the case study are limited to stations A and B. The period of each train run is indicated in Table 1.

Figure 2a shows the service intention including train lines and commercial dependencies between single train runs of each line. Table 1 below provides an example of constraints related to the hourly service of line 2 running from station A (St A) to station B (St B). Figure 2b shows the track infrastructure of the scenario together with the mesoscopic section topology indicating the section capacities by the corresponding number of horizontal lines.

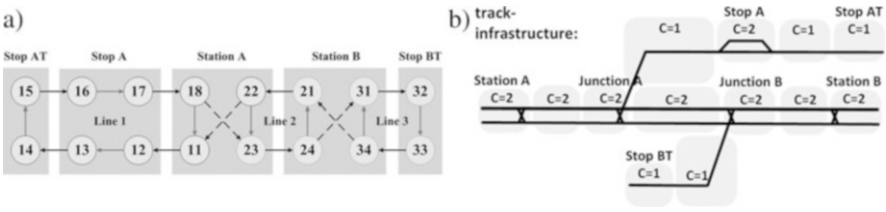


Fig. 2 (**a**) Schedule activity network with commercial dependencies modified from Goverde. Line 1, serving Stop A and Stop AT and connecting to Line 2 in Station A. Line 2 connecting Stations A and B. Line 3, serving Stop BT and connecting to Line 2 in Station B. (**b**) Track infrastructure of the test scenario with an indication of track capacities at each operation point. Operation points indicated as grey shaded boxes

Table 1 Line services with minimal trip times and periods

Line ID	Service ID	Minimum trip time	Period	Line ID	Service ID	Minimum trip time	Period
1	11	50	60	2	23	50	60
1	12	50	60	2	24	50	60
1	13	50	60	3	31	20	60
1	14	50	60	3	32	20	60
2	21	50	60	3	33	20	60
2	22	50	60	3	34	20	60

Odd numbers indicate train runs in one direction; even numbers indicate train runs of the same line in the opposite direction

The SI of test case A offers an hourly service of line 2 between major Stations A and B with connections to and from line 1 in station A and to and from line 3 in station B. A complete rotation of line 1 and 2 takes 120 min, one of line 3 takes 60 min. Therefore two vehicles are needed for rotations of line 1 and 2 and one is needed for line 3. Line services with train runs and corresponding periodicity and minimum circulation times are indicated in Table 1.

Figure 3 illustrates the results of the TCPESP algorithm for the given test scenario. In addition to the output of the conventional PESP algorithm given by arrival and departure event times, the result that we obtain from the TCPESP model includes track assignment information for each train run. The rail infrastructure of the test scenario consists of two single-track lines (line 1 and 3) and one double track line (line 2). We indicate the resulting track assignment by track numbers (Track 1 and Track 2) to each train run during run time on a given track section (see track diagram above each line diagram). There, the number of grey bold horizontal lines is identical to the number of tracks available at a corresponding operation points (Track 1 or both Track 1 and Track 2, respectively). From Fig. 3 it can be seen that the TCPESP algorithm only permits contra rotating train runs to meet in double track sections (line 1) and connecting train runs to meet in a station on neighboring tracks (platforms; St A: line 1 and 2, St B: line 2 and 3). Line styles correspond to directed train runs in both, the track diagrams and the time diagrams.

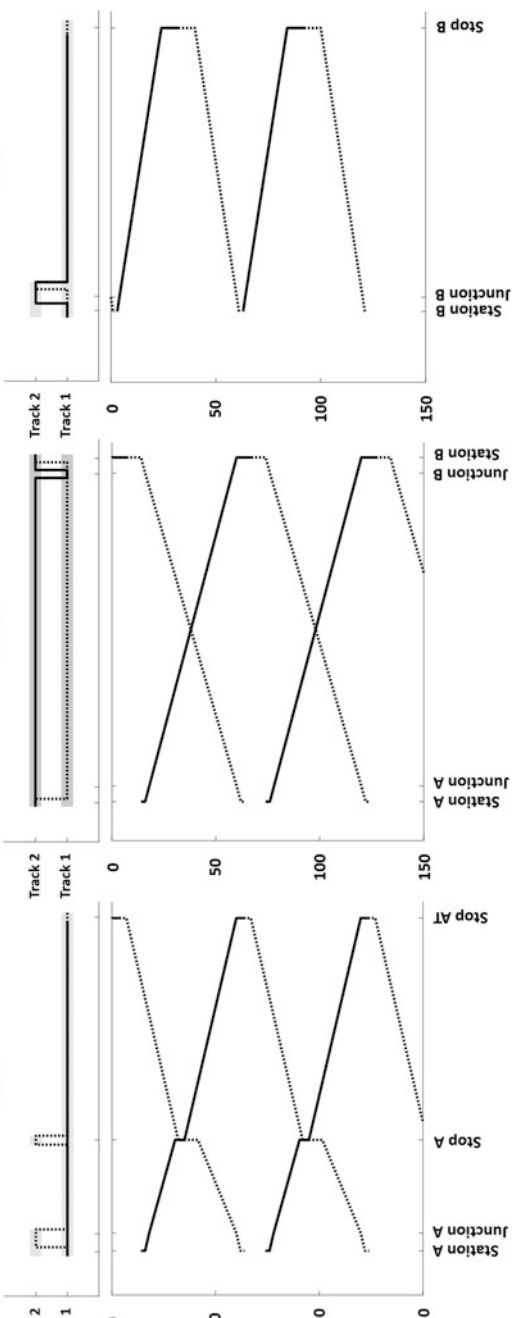


Fig. 3 Scheduling results obtained from of our TCPESP model. A train diagram with the arrival and departure event times is plotted together with the track assignment. Vertical axis: time between 0 and 150 min, horizontal axis: sequential locations. St A: station A, St B: station B, Stp A: Stop A, Stp AT: Final stop at AT. Track 1 and Track 2 with grey shaded horizontal lines above each location-time diagram indicate track assignments for each vehicle circulation of the three given lines

4 Discussion and Outlook

We introduced and successfully applied the new timetabling model TCPESP, which can be used to support timetable planners for generating train and vehicle schedules with track assignment. This model is based on an extension of the well-known PESP model and can be configured by using a standard schedule editor. Future developments include (i) the generation of the SI using a standard line planning method (see e.g [8]); (ii) the evaluation of timetable stability. In that way, we expect to further improve the quality of TCPESP results and contribute for speeding up and facilitating practical railway timetabling.

References

1. Hansen, I.A., Pachl, J. (eds.): Railway Timetable & Traffic: Analysis – Modelling – Simulation. Eurail Press, Hamburg (2008)
2. de Fabris, S., Longo, G., Medeossi, G., Pesenti, R.: Automatic generation of railway timetables based on a mesoscopic infrastructure model. J. Rail Transp. Plann. Manage. **4**, 2–13 (2014)
3. Bešinović, N., Goverde, R.M.P., Quaglietta, E., Roberti, R.: An integrated micro-macro approach to robust railway timetabling. Transp. Res. B Methodol. **87**, 14–32 (2016)
4. Caimi, G., Fuchsberger, M., Laumanns, M., Schüpbach, K.: Periodic railway timetabling with event flexibility. Networks. **57**(1), 3–18 (2011)
5. Liebchen, C., Möhring, R.H.: The modeling power of the periodic event scheduling problem: railway timetables – and beyond. In: Geraets, F., Kroon, L., Schoebel, A., Wagner, D., Zaroliagis, C. (eds.) Algorithmic Methods for Railway Optimization, Lecture Notes in Computer Science, vol. 4359, pp. 3–40. Springer, Berlin/Heidelberg (2007)
6. Caimi, G.: Algorithmic decision support for train scheduling in a large and highly utilised railway network. Diss. ETH Zürich Nr. 18581 (2009)
7. SMA: Viriato - software for railways. http://www.sma-partner.ch, Zurich. Accessed 01 July 2018
8. Friedrich, M., Hartl, M., Schiewe, A., Schöbel, A.: Integrating passengers' assignment in cost-optimal line planning. Technical Report 2017-5, Preprint-Reihe, INAM, Georg-August Universität Göttingen (2017)

Efficiency of Semi-autonomous Platooning Vehicles in High-Capacity Bus Services

Wei Zhang, Erik Jenelius, and Hugo Badia

1 Introduction

Several studies investigate the possibility of using fully (Level-5) autonomous vehicles as a supplement to current public transport system to serve personalized travel requests (e.g., autonomous taxis and autonomous on-demand bus services with flexible routes) [4, 6]. However, it is impractical to serve large demands with small vehicle capacities without fixed routes [8]. Therefore, some studies also consider replacing conventional buses with autonomous buses [3].

For high-demand scenarios, bus rapid transit (BRT) systems can deliver passengers efficiently on segregated roads with corresponding infrastructure. BRT can become a competitive public transport mode unless the demand is very low and sparsely distributed. Compared with traditional bus transit (in mixed traffic), BRT can avoid congestion to a large extent [7] .

Since fully autonomous buses are not yet ready to be implemented in practice due to safety concerns and low driving speed, this paper considers the adoption of semi-autonomous (Level-4 automation) buses as a more ready solution. Unlike fully autonomous buses, semi-autonomous buses need to operate as platoons[1] without drivers in the follower vehicles but with a driver in the leading vehicle.

[1] In this context, a platoon is a string of vehicles which drive closely to each other, with vehicle-to-vehicle communication and adaptive cruise control (ACC) or cooperative adaptive cruise control (CACC) technologies to maintain safety with short inter-vehicle distance. The number of vehicles contained in a platoon is called *platoon length*.

W. Zhang (✉) · E. Jenelius · H. Badia
Division of Transport Planning, KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: wzh@kth.se; wei.zhang@abe.kth.se

B. Fortz, M. Labbé (eds.), *Operations Research Proceedings 2018*,
Operations Research Proceedings, https://doi.org/10.1007/978-3-030-18500-8_72

By platooning, the labor cost is reduced, and the platoon capacity can adjust to demand without introducing significant operating cost fluctuations.

To compare the performance of semi-autonomous buses with conventional buses, we formulate the operating problem as a constrained optimization problem and solve it analytically. Numerical results are provided, with sensitivity analysis regarding different demand levels and capacity upper bounds.

2 Problem Formulation

Consider a corridor of length l where the hourly directional demand q is uniformly distributed along the line. The directional hourly demand at position x is $2q(lx - x^2)/l^2$. The maximum demand, $q/2$, appears at $l/2$.

The problem is to minimize the generalized cost C_{tot}, which is the sum of passengers' cost and the service provider's cost, by optimizing bus size s, platoon length N and service headway h. The passengers' cost includes access time cost C_{access}, waiting cost C_{wait} and riding cost C_{ride}, while the service provider's cost is composed of operating cost C_{oper} and capital cost C_{cptl}.

There are two types of buses, namely conventional buses ($i = \text{conv}$) and semi-autonomous buses ($i = \text{sa}$), which can be used in either traditional bus transit ($j = \text{bus}$) or BRT systems ($j = \text{BRT}$). The difference between traditional bus transit and BRT lies in the driving speed, stop spacing, and capital cost (mainly land cost and infrastructure cost). Assuming that the bus fleet is homogeneous, there are four possible combinations (i, j) of services in total.

2.1 Cost Components

Access Cost Given the distance between two consecutive bus stops d^j, the walking distance on average is $d^j/2$ for each user. The hourly access time cost for all users is:

$$C^j_{\text{access}} = c_{\text{access}} \frac{d^j}{2v_{\text{walk}}} 2q, \tag{1}$$

where c_{access} is the value of access time and v_{walk} is the walking speed.

Waiting Cost Assuming passengers arrive randomly to the stop, the average waiting time is half of the service headway h. Therefore, the hourly waiting cost is

$$C_{\text{wait}} = c_{\text{wait}} hq, \tag{2}$$

where c_{wait} is the value of waiting time.

Riding Cost The value of in-vehicle time is modeled as a linear function of the occupancy rate ϕ, $c_{\text{iv}}(\phi) = c_{\text{ride}} + c_{\text{dcf}}\phi$, where c_{dcf} measures the discomfort caused by crowding [5].

Integrating over the demand distribution along the line, the total riding cost is

$$C_{\text{ride}}^{j} = \frac{2ql}{v^{j}}\left(\frac{c_{\text{ride}}}{3} + \frac{2}{15}\frac{qhc_{\text{dcf}}}{Ns}\right), \tag{3}$$

where Ns is the capacity of the bus platoon, and v^{j} is the driving speed.

Operating Cost Vehicles are operated jointly in a bus platoon of $N \geq 1$ vehicles. To serve the demand with conventional buses, the hourly operating cost per vehicle is $c_{\text{oper}} + b_{\text{oper}}s$, where c_{oper} and b_{oper} are the fixed operating cost and marginal operating cost with respect to vehicle size s, respectively. For semi-autonomous buses, each platoon follower will experience a reduction η^{sa} in the labor cost. In general, the hourly operating cost is

$$C_{\text{oper}}^{ij} = \frac{2l\{[1 + (N-1)(1-\eta^{i})]c_{\text{oper}} + Nb_{\text{oper}}s\}}{v^{j}h}, \tag{4}$$

where $\eta^{\text{conv}} = 0$.

Capital Cost The capital cost includes infrastructure, land and rolling stock cost. Since BRT requires segregated lanes to ensure higher vehicle diving speed than traditional bus transit, we assume a fixed capital cost term c_0^{BRT} to capture the extra infrastructure and land needed by BRT. For conventional buses, the hourly capital cost per vehicle is $c_{\text{cptl}} + b_{\text{cptl}}s$, where c_{cptl} and b_{cptl} are the fixed capital cost and marginal capital cost with respect to vehicle size s, respectively. For semi-autonomous vehicles, there is an additional fixed capital cost β^{sa}. The hourly capital cost is:

$$C_{\text{cptl}}^{ij} = c_0^{j} + \frac{2lN[(1+\beta^{i})c_{\text{cptl}} + b_{\text{cptl}}s]}{v^{j}h}, \tag{5}$$

where $c_0^{\text{bus}} = 0$ and $\beta^{\text{conv}} = 0$.

2.2 Generalized Cost Minimization

For each of the four bus service scenarios, the cost minimization problem can be formulated as

$$\min_{h,N,s} C_{\text{tot}}^{ij} = C_{\text{access}}^{j} + C_{\text{wait}} + C_{\text{ride}}^{j} + C_{\text{oper}}^{ij} + C_{\text{cptl}}^{ij} \tag{6}$$

subject to

$$2Ns - hq \geq 0, \tag{7}$$

$$s_{\text{ub}} - s \geq 0, \tag{8}$$

$$N - 1 \geq 0, \tag{9}$$

where (7) ensures that the service is able to serve the maximum load, (8) limits the bus size to its upper bound and (9) ensures that there is at least one vehicle in the bus platoon.

We can divide the analysis into eight cases based on the KKT complementarity conditions, and obtain the optimal solution in three relevant cases:

Case 1: (7) and (8) are inactive while (9) is active gives

$$N = 1,\ s = \sqrt{\frac{4qlc_{\text{dcf}}[c_{\text{oper}} + (1+\beta^i)c_{\text{cptl}}]}{15v^j c_{\text{wait}}(b_{\text{oper}} + b_{\text{cptl}})}},\ h = \sqrt{\frac{2l[c_{\text{oper}} + (1+\beta^i)c_{\text{cptl}}]}{qv^j c_{\text{wait}}}}. \tag{10}$$

Case 2: (7) and (9) are inactive and (8) is active gives

$$N = \sqrt{\frac{4ql\eta^i c_{\text{oper}} c_{\text{dcf}}}{15c_{\text{wait}}v^j s_{\text{ub}}[(1-\eta^i)c_{\text{oper}} + (1+\beta^i)c_{\text{cptl}} + (b_{\text{oper}} + b_{\text{cptl}})s_{\text{ub}}]}},$$

$$s = s_{\text{ub}},\ h = \sqrt{\frac{2l\eta^i c_{\text{oper}}}{c_{\text{wait}}qv^j}}. \tag{11}$$

Case 3: (7) is inactive and (8) and (9) are active gives

$$N = 1,\ s = s_{\text{ub}},\ h = \sqrt{\frac{30ls_{\text{ub}}[c_{\text{oper}} + (1+\beta^i)c_{\text{cptl}} + (b_{\text{oper}} + b_{\text{cptl}})s_{\text{ub}}]}{15v^j s_{\text{ub}}c_{\text{wait}}q + 4q^2 l c_{\text{dcf}}}}. \tag{12}$$

3 Numerical Analysis

Studies show that the in-vehicle time cost increases by 50% if the bus is full [2]. Therefore, we use $c_{\text{dcf}} = c_{\text{ride}}/2$. Other parameters[2] are from [1, 3, 7]: $\beta^{\text{sa}} = 0.2$, $b_{\text{oper}} = 0.75$ SEK/h/vehicle, $b_{\text{cptl}} = 1.01$ SEK/h/vehicle, $c_0 = 60824$ SEK/h,

[2]Units are converted from AUD to SEK (1 AUD = 6.41 SEK).

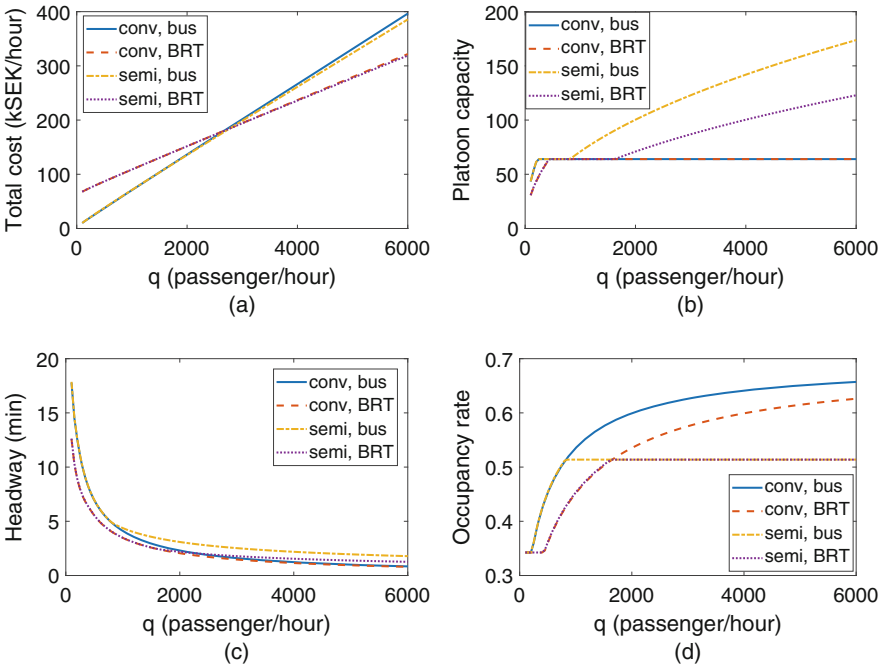


Fig. 1 Results with respect to different demand levels, $s_{\text{ub}} = 64$ pax. (**a**) Total cost with respect to demand level. (**b**) Platoon capacity with respect to demand level. (**c**) Headway with respect to demand level. (**d**) Occupancy rate with respect to demand level

$c_{\text{access}} = 66.1$ SEK/h, $c_{\text{wait}} = 79.35$ SEK/h, $c_{\text{ride}} = 56.28$ SEK/h, $c_{\text{dcf}} = 28.14$ SEK/h, $c_{\text{oper}} = 334.6$ SEK/h/vehicle, $c_{\text{cptl}} = 14.24$ SEK/h/vehicle, $d^{\text{BRT}} = 0.8$ km, $d^{\text{bus}} = 0.4$ km, $l = 15$ km, $v_{\text{walk}} = 4$ km/h, $v^{\text{BRT}} = 30$ km/h, $v^{\text{bus}} = 15$ km/h.

Results with respect to different demand levels q are shown in Fig. 1. As demand increases, the optimal bus size increases until the upper bound s_{ub} is reached. The capacity of semi-autonomous bus platoons may exceed the upper vehicle size bound due to platooning (Fig. 1b). The headways in all four scenarios decrease as q increases (Fig. 1c). For low demand levels, BRT provides shorter headways than traditional bus transit, e.g., 5 min when $q = 100$ pax/h. For high demand levels, semi-autonomous buses yield slightly longer headways (within 1 min given $q = 6000$ pax/h) than conventional buses in both BRT and bus transit.

The occupancy rate of conventional buses increases as q increases (except when $q < 400$ pax/h), while for semi-autonomous buses, it is fixed to 0.51 beyond 850 and 1650 pax/h, for bus and BRT, respectively (Fig. 1d). This indicates that semi-autonomous buses may offer a better riding experience when q is high. The total cost savings by using semi-autonomous bus can be up to 10.8 kSEK/h in bus transit and 2.8 kSEK/h in BRT when $q = 6000$ pax/h (Fig. 1a). Semi-autonomous buses reduce both passengers' costs and the service provider's costs. However, semi-

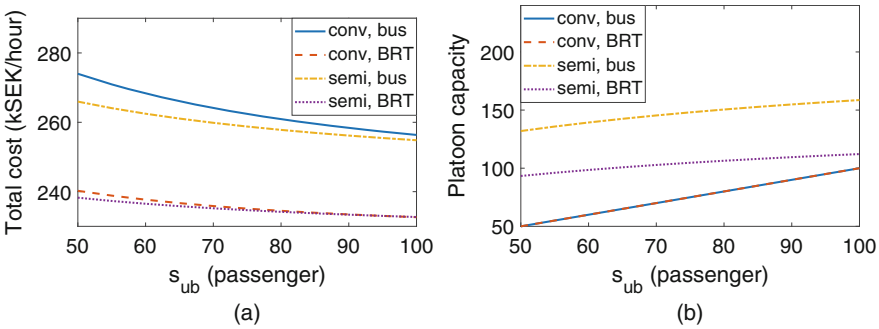


Fig. 2 Results with respect to different upper bound s_{ub}, $q = 4000$ pax/h. (**a**) Total cost with respect to bus size upper bound. (**b**) Platoon capacity with respect to bus size upper bound

autonomous buses start to lose advantage when s_{ub} becomes larger (see Fig. 2). For example, given $q = 4000$ pax/h, semi-autonomous buses in BRT can save 2.0 kSEK/h when $s_{\text{ub}} = 50$, but are 51.2 SEK/h more expensive when $s_{\text{ub}} = 100$.

4 Conclusion

The study shows that semi-autonomous bus can be implemented to serve medium and high demands, in traditional bus transit and BRT respectively. The optimal service mode to use is dependent on the specific demand level and the bus capacity upper bound. If large buses are not allowed in the transit network, semi-autonomous buses can be competitive. Conventional buses will eventually fail to serve very high demand since solely adjusting service headways will lead to unachievable short headways and extremely large fleet size, which creates heavy congestion and other externalities.

Acknowledgements The support from China Scholarship Council is gratefully acknowledged.

References

1. Australian Transport Council: National guidelines for transport system management in Australia. https://www.atap.gov.au/technical-support-library/ngtsm/files/National_Guidelines_Volume_4.pdf (2006). Accessed 15 July 2018
2. Börjesson, M., Fung, C.M., Proost, S.: Optimal prices and frequencies for buses in Stockholm. Econ. Transp. **9**, 20–36 (2017)
3. Bösch, P.M., Becker, F., Becker, H., Axhausen, K.W.: Cost-based analysis of autonomous mobility services. Transp. Policy **64**, 76–91 (2018)
4. Fagnant, D.J., Kockelman, K.M., Bansal, P.: Operations of shared autonomous vehicle fleet for Austin, Texas, market. Transp. Res. Rec. J. Transp. Res. Board **2536**(2536), 98–106 (2015)

5. Jara-Díaz, S., Gschwender, A.: Towards a general microeconomic model for the operation of public transport. Transp. Rev. **23**(4), 453–469 (2003)
6. Scheltes, A., de Almeida Correia, G.H.: Exploring the use of automated vehicles as last mile connection of train trips through an agent-based simulation model: an application to Delft, Netherlands. Int. J. Transp. Sci. Technol. **6**(1), 28–41 (2017)
7. Tirachini, A., Hensher, D.A., Jara-Díaz, S.R.: Restating modal investment priority with an improved model for public transport analysis. Transport. Res. E-Log **46**(6), 1148–1168 (2010)
8. White, P.: The roles of conventional and demand-responsive bus services. In: Paratransit: Shaping the Flexible Transport Future, pp. 307–330. Emerald Group Publishing Limited, Bingley (2016)