Logic-Based Decomposition Methods for the Travelling Purchaser Problem

Kyle E.C. Booth[(✉)], Tony T. Tran, and J. Christopher Beck

Department of Mechanical and Industrial Engineering,
University of Toronto, Toronto, ON M5S 3G8, Canada
{kbooth,tran,jcb}@mie.utoronto.ca

Abstract. We present novel branch-and-check and logic-based Benders decomposition techniques for the Travelling Purchaser Problem, an important optimization problem with applications in vehicle routing, logistics, and warehouse management. Our master problem determines a set of markets and directed travel arcs that satisfy product purchase constraints with relaxed travel costs. Our subproblem identifies subtours within this master assignment and produces a set of generalized subtour elimination cuts. We show that the proposed technique demonstrates strong performance on the asymmetric problem variants, finding optimal solutions to previously unsolved instances, while performing competitively on a number of symmetric problem classes. Furthermore, our model is implemented unchanged for the four problem variants whereas other state-of-the-art approaches are variant-specific.

1 Introduction

Given a set of markets, each with a set of available products, the Travelling Purchaser Problem (TPP) aims to determine a simple cycle among a subset of markets that minimizes the sum of travel cost and purchase cost for the set of products required by the traveller. The problem is $\mathcal{NP}$-Hard [23] and generalizes both the Travelling Salesman Problem (TSP) [10] and the Uncapacitated Facility Location Problem (UFLP) [9].

Our primary contribution is the development of an exact decomposition model to solve the TPP. The decomposition uses mixed-integer programming for the master problem and a straightforward subtour identification algorithm to generate cuts. The method is simple to implement in commercially available solvers, and does not require sophisticated separation procedures, nor an in-depth polytope analysis [19,24]. To our knowledge, there is no other single approach that has been used without modification to efficiently solve the uncapacitated, capacitated, asymmetric, and symmetric problem variants. Our approach achieves strong performance on both the capacitated and uncapacitated asymmetric instances while remaining competitive on symmetric problems.

As far as we are aware, this work is the first application of branch-and-check and logic-based Benders decomposition (LBBD) for the TPP, though an LBBD-inspired heuristic approach has been investigated [6] and served as an initial inspiration for this work.

2   Background

In this section we define the TPP and review existing relevant work, focusing on exact algorithms rather than heuristic approaches (e.g., [5,12,13]).

2.1   Problem Definition

Following Laporte et al. [19], consider a set of markets $M := \{v_0, v_1, ..., v_n\}$, where v_0 is a depot, and a set of available products $K := \{p_1, ..., p_m\}$. The demand for product p_k, the quantity of product that must be purchased, is d_k and the price of p_k at v_i is b_{ki}. Each product, p_k, can be purchased at a subset of the markets, M_k, and the quantity of p_k available at v_i is q_{ki}. We define $M^* := \{v_0\} \cup \{v_i \in M : \exists\ p_k \in K \text{ such that } \sum_{v_j \in M_k \setminus \{v_i\}} q_{kj} < d_k\}$ as the set of required markets. The travel cost between markets v_i and v_j is c_{ij}. We must find a simple cycle among a subset of markets such that all product demand is satisfied and the sum of the travel and product purchase costs are minimized.

  We present a mixed-integer programming (MIP) model in Fig. 1. The model is based on Laporte et al. [19] but uses the lifted Miller-Tucker-Zemlin (MTZ) subtour elimination formulation [8,21]. The decision variables are:

  $z_i := 1$ if market v_i is visited and 0 otherwise
  $x_{ij} := 1$ if market v_i is visited directly before v_j and 0 otherwise
  $y_{ki} :=$ the purchased quantity of p_k at market v_i
  $u_i :=$ a positive variable used for MTZ subtour elimination [21]

  Equation (1) minimizes the sum of travel and product purchase costs. Constraints (2) and (3) represent the degree constraints for each market. Constraint (4) ensures that the demand for each product is satisfied while Constraint (5) ensures that quantity of a product purchased at a market is contingent on both the decision to visit that market and the product quantity available. Constraint (6) represents the lifted MTZ [8,21] formulation for subtour elimination. Constraint (8) ensures that z_i is set to a value of 1 if market v_i is a required market.

2.2   Problem Variants

The majority of TPP variants addressed in the literature fall along two dimensions: capacitated vs. uncapacitated, and symmetric vs. asymmetric. A problem is uncapacitated if each market sells enough of its products to satisfy the traveller's demand for those products (i.e., $q_{ki} \geq d_k, \forall v_i \in M_k, p_k \in K$) and, therefore, a given product is always satisfied by a single market. In capacitated problems, the quantity of a product at a market may or may not completely satisfy the demand (i.e., $0 < q_{ki} \leq d_k, \forall v_i \in M_k$) and so the traveller may have to visit several markets to satisfy the demand for a single product. A problem is symmetric if $c_{ij} = c_{ji}$ holds, and asymmetric if it does not. These dimensions combine to form four problem variations: uncapacitated-symmetric (U-STPP), uncapacitated-asymmetric (U-ATPP), capacitated-symmetric (C-STPP), and capacitated-asymmetric (C-ATPP).

$$\min \sum_{v_i \in M} \sum_{v_j \in M, v_j \neq v_i} c_{ij} x_{ij} + \sum_{p_k \in K} \sum_{v_i \in M_k} b_{ki} y_{ki} \tag{1}$$

$$\text{s.t.} \quad \sum_{v_i \in M, v_i \neq v_j} x_{ij} = z_j \qquad\qquad \forall\, v_j \in M, \tag{2}$$

$$\sum_{v_j \in M, v_j \neq v_i} x_{ij} = z_i \qquad\qquad \forall\, v_i \in M, \tag{3}$$

$$\sum_{v_i \in M_k} y_{ki} = d_k \qquad\qquad \forall\, p_k \in K, \tag{4}$$

$$y_{ki} \leq z_i q_{ki} \qquad\qquad \forall\, p_k \in K; v_i \in M_k, \tag{5}$$

$$u_i - u_j + (|M| - 1)x_{ij} + (|M| - 3)x_{ji} \leq |M| - 2 \quad \forall\, v_i, v_j \geq v_1; v_i \neq v_j, \tag{6}$$

$$1 \leq u_i \leq |M| - 1 \qquad\qquad \forall\, v_i \geq v_1, \tag{7}$$

$$z_i = 1 \qquad\qquad \forall\, v_i \in M^*, \tag{8}$$

$$z_i \in \{0, 1\} \qquad\qquad \forall\, v_i \in M \setminus M^*, \tag{9}$$

$$x_{ij} \in \{0, 1\} \qquad\qquad \forall\, v_i, v_j \in M; v_i \neq v_j, \tag{10}$$

$$0 \leq y_{ki} \leq d_k \qquad\qquad \forall\, p_k \in K; v_i \in M_k. \tag{11}$$

Fig. 1. A MIP Model for the Travelling Purchaser Problem based on Laporte et al. [19] with lifted MTZ subtour elimination constraints [8,21].

The MIP model in Fig. 1 is valid for all four problem variants as the differences are embodied in the instance data: the symmetricity difference is due to the travel cost matrix data and the uncapacitated variants simply assign $d_k = 1, \forall p_k \in K$ and $q_{ki} = 1, \forall p_k \in K; v_i \in M_k$.

2.3 Related Work

The first exact approach to the TPP was a lexicographic search algorithm capable of solving the U-STPP and U-ATPP instances with $|M| = 12$ and $|K| = 10$ (12×10) [23]. Singh et al. [25] proposed a branch-and-bound method for the U-STPP and U-ATPP that utilized the relaxation of UFLP constraints to generate lower bounds. The approach solved asymmetric instances of size 25×50 and symmetric instances of size 25×30. Laporte et al. [19] proposed the first capacitated formulation of the symmetric TPP and developed a branch-and-cut approach for the U-STPP and C-STPP. This method solved instances of size 200×200 and remains a state of the art for symmetric variants. Riera et al. [24] developed a state-of-the-art branch-and-cut approach for the U-ATPP and C-ATPP, solving instances of 200×200. More recently, Cambazard et al. [7] developed a constraint programming approach for the U-STPP, solving instances of size 250×200, often out-performing Laporte et al.

3 LBBD and Branch-and-Check for the TPP

We investigate both logic-based Benders decomposition (LBBD) and branch-and-check (B&C), with the notion that the TPP can benefit from the delayed enforcement of certain constraints. The decomposition structure is the same for both

approaches, the difference concerning whether the sub-problem is solved at optimal or feasible solutions to the master problem. LBBD [15] uses logic-based sub-problems to produce valid Benders cuts [11] for the master problem. In LBBD, the master problem is solved to optimality and the solution to this relaxed problem is utilized to solve the subproblem(s) and generate cuts. The master problem is then re-solved, and this iterative process continues until the solution to the master problem, with all generated cuts, is valid with respect to the subproblems, and thus is a globally optimal solution. B&C [26] is a variation of LBBD where the subproblem(s) are solved whenever a feasible solution is found during the branch-and-bound search of the master problem. Problems with more difficult master problems, as compared to the subproblems, are more suited for branch-and-check, whereas difficult subproblems comparatively favor LBBD [4].

Assignment Master Problem. In the decomposition proposed, the master problem is a relaxation of the full MIP model (Fig. 1) through omission of the subtour elimination constraints and associated variables: that is, the removal of Constraints (6) and (7). A solution to the master problem consists of an integer set of assigned markets, z_i, and directed travel arcs, x_{ij}, that satisfy product purchase requirements while allowing subtours. It is natural, therefore, for the subproblem to identify subtours and eliminate them through cut generation.

Subtour Identification Subproblem. Our approach consists of identifying these subtours, evaluating their candidacy as globally feasible solutions, and producing generalized subtour elimination [16] cuts when appropriate.

Due to Constraints (2) and (3), the master solution consists of a set of one or more disjoint tours of the selected markets. Since our master assignments are integer, a trivial depth first search is sufficient to identify the unique set of subtours, S^h, in the hth master problem solution where each $s_\ell^h \in S^h$ consists of a set of markets in a subtour.

For each subtour we first assess whether it is, by itself, a feasible solution to the global problem; that is, does s_ℓ^h satisfy all product purchase requirements and include the depot, $v_0 \in s_\ell^h$. While such subtours will not exist for LBBD, due to the optimality of the master problem solution, for B&C at most one such subtour may exist per iteration.[1] If such a subtour, $\hat{s}^h$, exists, we remove it from S^h and use it as a new global incumbent solution. At the same time, for each subtour $s_\ell^h \in \bar{S}^h := S^h \setminus \{\hat{s}^h\}$, we introduce a generalized subtour elimination cut defined as follows:

$$\sum_{v_i \in s_\ell^h} \sum_{v_j \in s_\ell^h, v_j \neq v_i} x_{ij} \leq \sum_{v_i \in s_\ell^h} z_i + \psi_{s_\ell^h} - 1 \qquad \forall s_\ell^h \in \bar{S}^h, \tag{12}$$

$$\psi_{s_\ell^h} + z_i \leq 1 \qquad\qquad \forall v_i \in s_\ell^h; s_\ell^h \in \bar{S}^h, \tag{13}$$

$$\psi_{s_\ell^h} \in \{0, 1\} \qquad\qquad \forall s_\ell^h \in \bar{S}^h. \tag{14}$$

---

[1] The limit of one per iteration is due to the depot inclusion condition.

The left hand side of Cut (12) is the number of chosen arcs in the (complete) sub-graph induced by the markets in subtour s_ℓ^h. The right hand side defines an upper bound for this value, prohibiting the creation of a subtour of any permutation of the markets as well as removing some, but not all, subtours among subsets of the markets. The subset prohibitions are achieved through the use of the $\sum_{v_i \in s_\ell^h} z_i$ term instead of simply the cardinality of the markets, $|s_\ell^h|$.

The cut adds of a new auxiliary variable, $\psi_{s_\ell^h}$, for each $s_\ell^h \in \bar{S}^h$ which, due to (12), is constrained to the value 1 if none of the markets in the subtour are chosen in a subsequent iteration (i.e., if $\sum_{v_i \in s_\ell^h} z_i = 0$) and, due to (13), is constrained to 0 otherwise. Functionally, $\psi_{s_\ell^h}$ ensures the global validity of the cut. Without its inclusion, if none of the markets $v_i \in s_\ell^h$ were chosen in a subsequent iteration, Cut (12) would reduce to $0 \leq 0 - 1$, removing a potentially globally optimal solution.

The validity of the cut is easily seen. Each cut eliminates at least one subtour from the master solution space and as the removed subtour does not itself constitute a globally feasible solution, no such solutions are removed. Convergence to optimality is then based on the finite (though large) number of subtours.

This cut has a similar purpose to the one proposed for the Orienteering Problem [14,16], though is different through the use of variable generation. Initial experimentation suggests that our cut performs more effectively, in general, than the one proposed in Laporte et al. [16], though this is an area we intend to explore more thoroughly. We note that an equivalent generalized connectivity cut [16] can be used as well, using the cut-set of directed arcs.

In traditional cut-based approaches for cycle problems, subtour constraints and integrality requirements are relaxed and violated inequalities are separated based on fractional solutions of the resulting linear program (LP). When the LP is solved, a max-flow (or min-cut) problem is solved to identify and separate violated tour constraints [22]. Additional cutting planes are available, most notably the class of comb inequalities [2]. This standard approach relies heavily on the performance of the LP solver, requiring active management of model size due to the large number of valid inequalities introduced via the various separation procedures. Conversely, subtour elimination based on integer assignments has been applied to the TSP [18,20], but not to our knowledge the TPP.

4   Computational Results

In this section we present benchmark results of the proposed LBBD and B&C formulations on the four main variants of the TPP.

4.1   Benchmark Problems

The instance set we use is well-established in the literature [19,24] and consists of 745 instances across six problems classes. The capacitated instances introduce a parameter λ that dictates how traveller demand, d_k, relates to the available

quantity of a product, q_{ki}: $d_k := \lceil \lambda \max_{v_i \in M_k}(q_{ki}) + (1 - \lambda) \sum_{v_i \in M_k} q_{ki} \rceil$ [19]. Each instance set consists of five instances for each combination of $|M|$, $|K|$ and λ considered.

Class 1. U-STPP [19] where $|M| = 33$, $|K| \in \{50, 100, 150, 200, 250\}$, for a total of $\mathcal{N} = 25$ problem instances.
Class 2 and 3. U-STPP [19] where $|M| \in \{50, 100, 150, 200\}$, $|K| \in \{50, 100, 150, 200\}$, for a total of $\mathcal{N} = 80$ problem instances in each.
Class 4. C-STPP [19] where $|M| \in \{50, 100, 150, 200\}$, $|K| \in \{50, 100, 150, 200\}$, and $\lambda \in \{0.5, 0.7, 0.9, 0.99\}$, for a total of $\mathcal{N} = 280$ problem instances.
Class 1A. U-ATPP [24] where $|M| \in \{50, 100, 150, 200\}$, $|K| \in \{50, 100, 150, 200\}$, with a total of $\mathcal{N} = 80$ problem instances.
Class 2A. C-ATPP [24] where $|M| \in \{50, 100\}$, $|K| \in \{50, 100, 150, 200\}$, and $\lambda \in \{0.5, 0.8, 0.9, 0.95, 0.99\}$ with a total of $\mathcal{N} = 200$ instances.

4.2 Experimental Details

We implement our methods with the CPLEX 12.6.2 mixed-integer programming solver in C++. For B&C, we utilize lazy constraints to trigger subproblem solving and cut generation whenever a feasible master solution is found. As CPLEX does not directly support variable generation within branch-and-bound, we pre-allocate a number of $\psi_{s_\ell^h}$ variables which the cuts then make use of. This situation is not ideal, as it leads to a master problem with variables that may never be utilized. Better B&C performance is likely with true variable generation.[2] For the LBBD approach, since each master iteration is solved anew, we do not require in-search variable generation and do not suffer the same B&C limitations.

We compare to published results from the aforementioned state-of-the-art methods. As the four problem variants have never been approached in one study, we adapt our run-times to be appropriate for different experimental designs in the literature. We use a run-time limit of 3,600 s for symmetric experiments, whereas previous papers use longer run-times (7,200 and 18,000 s). For asymmetric instances, we use a run-time limit of 7,200 s to match the limits in the existing papers.

Our experiments use a Xeon 3.5 GHz processor machine with 16 GB of RAM running OS X Yosemite. Laporte et al. and Riera et al. utilize much older Pentium 500 MHz and AMD 1.33 GHz machines, respectively, running Linux with CPLEX 6.0. The Cambazard paper uses a Xeon 2.66 GHz processor machine with 16 GB of RAM running Linux 2.6.25 x64 and customized CP software. Due to the mix of software and hardware, the results should be interpreted with care.

4.3 Results

Table 1 presents the results for the asymmetric variants (Classes 1A and 2A) where the branch-and-cut approaches of Riera et al. [24] represent the state of

---

[2] Experiments using dynamic variable creation in SCIP [1] show a relative improvement in B&C compared to LBBD though both CPLEX implementations are faster.

the art. Two methods are presented in Riera et al.: Riera$_{B\&CUT}$ that uses a customized branch-and-cut algorithm with sophisticated separation procedures, and Riera$_{TRANS}$, that transforms the ATPP into its symmetric counterpart, and then uses the branch-and-cut of Laporte et al. [19].

The results for asymmetric instances with our approach show speed-up factors of 3 to 70 compared to the previous state of the art, including solving a number of previously unsolved instances. We suspect the reason for this superior performance is rooted within our master problem relaxation, namely the market assignment relaxation for the TPP. As demonstrated by Balas et al. [3], the assignment problem (AP) relaxation for the TSP is very strong when the cost matrix, c_{ij}, is generated randomly based on a uniform distribution, and thus asymmetric, resulting in near-optimal solutions. Since the instances for Class 1A and 2A are generated randomly based on a uniform distribution, it would appear this property holds for the TPP as well. Again, hardware and software differences make this comparison less clear cut, though we believe they do provide supporting evidence for our techniques as strong contenders for the new state of the art on asymmetric TPP problems.

Table 2 presents the results on the symmetric instances (Classes 1–4), compared to the state-of-the-art approaches. Laporte$_{B\&CUT}$ [19] utilizes branch-and-cut with valid inequalities to strengthen the linear relaxation with sophisticated separation techniques. The Cambazard [7] approach, Cambazard$_{CP-PM}$, uses a constraint programming (CP) model with a p-median constraint, originally intended for solving TPP problems with a bounded number of visits.

Our proposed decomposition techniques are competitive with the existing state of the art on Classes 1 and 2 while falling short on instances of Class 3 and 4 for larger values of $|M|$. For the symmetric case, our master assignment relaxation is much weaker [3] for the underlying cycle problem, which tends to contain many more subtours of size 2 [17] than the asymmetric counterparts, resulting in excessive computation time for their elimination.

Table 1. Asymmetric results vs. the state of the art. $\mathcal{C}$ is the problem class and $\mathcal{N}$ the number of instances for each value of $|M|$. #F indicates number of instances that were not proved optimal in 7,200 second limit. Run-times are arithmetic mean CPU values.

Asymmetric problems (U-ATPP, C-ATPP)										
Problem			B&C		LBBD		Riera$_{B\&CUT}$		Riera$_{TRANS}$	
$\mathcal{C}$	$\mathcal{N}$	$|M|$	Avg (s)	#F	Avg (s)	#F	Avg (s)	#F	Avg (s)	#F
1A	20	50	1.5	0	2.3	0	14.0	0	33.8	0
	20	100	24.8	0	67.7	0	1,083.0	0	1,076.9	2
	20	150	166.6	0	485.2	1	1,697.6	4	2,779.1	4
	20	200	731.9	1	1,147.1	2	3,331.9	11	3,634.2	11
2A	100	50	1.7	0	2.8	0	100.5	0	152.4	0
	100	100	34.9	0	86.7	0	2,557.9	24	2,397.0	36

Table 2. Symmetric results vs. the state of the art. Notation is identical to Table 1 except with a 3,600 second run-time limit. '−' indicates the method was not attempted.

Symmetric problems (U-STPP, C-STPP)										
Problem			B&C		LBBD		Laporte$_{B\&CUT}$		Cambazard$_{CP-PM}$	
$\mathcal{C}$	$\mathcal{N}$	$|M|$	Avg (s)	#F	Avg (s)	#F	Avg (s)	#F	Avg (s)	#F
1	25	33	6.0	0	85.5	0	28.0	0	−	−
2	20	50	0.9	0	1.1	0	1.9	0	−	−
	20	100	9.6	0	6.9	0	12.2	0	−	−
	20	150	62.4	0	81.5	0	140.4	0	−	−
	20	200	221.1	0	280.0	0	298.5	1	−	−
3	20	50	11.8	0	270.5	0	23.9	0	7.1	0
	20	100	1,872.5	8	2,908.7	15	299.8	0	528.6	2
	20	150	3,244.6	17	3,513.8	19	1,725.1	1	821.94	2
	20	200	3,210.1	16	3,158.5	17	2,621.0	11	1,383.7	7
4	80	50	9.2	0	113.5	0	23.1	0	−	−
	80	100	677.7	8	1,174.2	19	402.3	2	−	−
	80	150	1,603.1	24	1,594.4	31	1,281.2	16	−	−
	40	200	3,020.6	29	2,084.9	20	2,414.6	19	−	−

5   Conclusions

We presented strategies for solving the Travelling Purchaser Problem with branch-and-check and logic-based Benders decomposition. We utilize a MIP model for the master problem, determining a set of markets that satisfy product purchase costs while relaxing the tour requirement. Our subproblem produces generalized subtour elimination cuts with variable generation.

Numerical results indicate strong performance on the asymmetric problem variants (both capacitated and uncapacitated) with order of magnitude speed-ups observed, albeit with differing hardware and software. On the symmetric instances, the performance was weaker, achieving about the same performance as the existing state of the art (on older hardware) on some problem classes but not achieving the same performance on others.

Notably, our model is applicable without modification across all four of the primary variants of the TPP while the current state-of-the-art techniques are variant-specific, exploiting sophisticated valid inequalities, separation schemes, and polytope analyses.

We plan to investigate algorithm extensions including primal solution heuristics, alternate cuts (e.g., [27]), as well as to perform a deeper analysis into the impact of symmetricity in order to improve performance on symmetric instances. We believe that the methods presented in this paper can be applied to more complex TPP-variants, as well as other routing problems.

References

1. Achterberg, T.: SCIP: solving constraint integer programs. Math. Program. Comput. **1**(1), 1–41 (2009)
2. Applegate, D., Bixby, R., Cook, W., Chvátal, V.: On the solution of traveling salesman problems. Rheinische Friedrich-Wilhelms-Universität Bonn (1998)
3. Balas, E., Toth, P.: Branch and bound methods for the traveling salesman problem. Technical report MSRR-488, DTIC Document (1983)
4. Beck, J.C.: Checking-up on branch-and-check. In: Cohen, D. (ed.) CP 2010. LNCS, vol. 6308, pp. 84–98. Springer, Heidelberg (2010)
5. Bontoux, B., Feillet, D.: Ant colony optimization for the traveling purchaser problem. Comput. Oper. Res. **35**(2), 628–637 (2008)
6. Burt, C.N., Lipovetzky, N., Pearce, A.R., Stuckey, P.J.: Approximate unidirectional benders decomposition. In: Proceedings of PlanSOpt-15 Workshop on Planning, Search and Optimization AAAI-15 (2015)
7. Cambazard, H., Penz, B.: A constraint programming approach for the traveling purchaser problem. In: Milano, M. (ed.) CP 2012. LNCS, vol. 7514, pp. 735–749. Springer, Heidelberg (2012)
8. Desrochers, M., Laporte, G.: Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. Oper. Res. Lett. **10**(1), 27–36 (1991)
9. Erlenkotter, D.: A dual-based procedure for uncapacitated facility location. Oper. Res. **26**(6), 992–1009 (1978)
10. Flood, M.M.: The traveling-salesman problem. Oper. Res. **4**(1), 61–75 (1956)
11. Geoffrion, A.M.: Generalized benders decomposition. J. Optim. Theor. Appl. **10**(4), 237–260 (1972)
12. Goerler, A., Schulte, F., Voß, S.: An application of late acceptance hill-climbing to the traveling purchaser problem. In: Pacino, D., Voß, S., Jensen, R.M. (eds.) ICCL 2013. LNCS, vol. 8197, pp. 173–183. Springer, Heidelberg (2013)
13. Goldbarg, M.C., Bagi, L.B., Goldbarg, E.F.G.: Transgenetic algorithm for the traveling purchaser problem. Eur. J. Oper. Res. **199**(1), 36–45 (2009)
14. Golden, B.L., Levy, L., Vohra, R.: The orienteering problem. Naval Res. Logistics (NRL) **34**(3), 307–318 (1987)
15. Hooker, J.N., Ottosson, G.: Logic-based benders decomposition. Math. Program. **96**(1), 33–60 (2003)
16. Laporte, G.: Generalized subtour elimination constraints and connectivity constraints. J. Oper. Res. Soc. **37**, 509–514 (1986)
17. Laporte, G.: The traveling salesman problem: an overview of exact and approximate algorithms. Eur. J. Oper. Res. **59**(2), 231–247 (1992)
18. Laporte, G., Nobert, Y.: A cutting planes algorithm for the m-salesmen problem. J. Oper. Res. Soc. **31**, 1017–1023 (1980)
19. Laporte, G., Riera-Ledesma, J., Salazar-González, J.-J.: A branch-and-cut algorithm for the undirected traveling purchaser problem. Oper. Res. **51**(6), 940–951 (2003)
20. Miliotis, P.: Integer programming approaches to the travelling salesman problem. Math. Program. **10**(1), 367–378 (1976)
21. Miller, C.E., Tucker, A.W., Zemlin, R.A.: Integer programming formulation of traveling salesman problems. J. ACM (JACM) **7**(4), 326–329 (1960)
22. Padberg, M., Rinaldi, G.: A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. SIAM Rev. **33**(1), 60–100 (1991)
23. Ramesh, T.: Traveling purchaser problem. Opsearch **18**(1–3), 78–91 (1981)

24. Riera-Ledesma, J., Salazar-González, J.-J.: Solving the asymmetric traveling purchaser problem. Ann. Oper. Res. **144**(1), 83–97 (2006)
25. Singh, K.N., van Oudheusden, D.L.: A branch and bound algorithm for the traveling purchaser problem. Eur. J. Oper. Res. **97**(3), 571–579 (1997)
26. Thorsteinsson, E.S.: Branch-and-check: a hybrid framework integrating mixed integer programming and constraint logic programming. In: Walsh, T. (ed.) CP 2001. LNCS, vol. 2239, pp. 16–30. Springer, Heidelberg (2001)
27. Tran, T.T., Araujo, A., Beck, J.C.: Decomposition methods for the parallel machine scheduling problem with setups. INFORMS J. Comput. **28**(1), 83–95 (2016)