Distribution with Quality of Service Considerations

Orlis, Christos; Laganá, Demetrio; Dullaert, Wout; Vigo, Daniele

published in
Omega (United Kingdom)
2020

DOI (link to publisher)
10.1016/j.omega.2019.02.003

document version
Publisher's PDF, also known as Version of record

document license
Article 25fa Dutch Copyright Act

Omega 93 (2020) 102034

Contents lists available at ScienceDirect

Omega

journal homepage: www.elsevier.com/locate/omega

Distribution with Quality of Service Considerations: The Capacitated Routing Problem with Profits and Service Level Requirements☆

Christos Orlis[a,*], Demetrio Laganá[b], Wout Dullaert[a], Daniele Vigo[c,a]

[a] Department of Supply Chain Analytics, Vrije Universiteit Amsterdam, The Netherlands
[b] Department of Mechanical, Energy and Management Engineering, University of Calabria, Italy
[c] Department of Electrical, Electronic and Information Engineering, University of Bologna, Italy

ARTICLE INFO

Article history:
Received 19 April 2018
Accepted 6 February 2019

Keywords:
Routing with profits
Cash logistics
ATM cash replenishment
Service level requirements
Branch-and-Cut

ABSTRACT

Inspired by a problem arising in cash logistics, we propose the Capacitated Routing Problem with Profits and Service Level Requirements (CRPPSLR). The CRPPSLR extends the class of Routing Problems with Profits by considering customers requesting deliveries to their (possibly multiple) service points. Moreover, each customer imposes a service level requirement specifying a minimum-acceptable bound on the fraction of its service points being delivered. A customer-specific financial penalty is incurred by the logistics service provider when this requirement is not met. The CRPPSLR consists in finding vehicle routes maximizing the difference between the collected revenues and the incurred transportation and penalty costs in such a way that vehicle capacity and route duration constraints are met. A fleet of homogeneous vehicles is available for serving the customers. We design a branch-and-cut algorithm and evaluate the usefulness of valid inequalities that have been effectively used for the capacitated vehicle routing problem and, more recently, for other routing problems with profits. A real-life case study taken from the cash supply chain in the Netherlands highlights the relevance of the problem under consideration. Computational results illustrate the performance of the proposed solution approach under different input parameter settings for the synthetic instances. For instances of real-life problems, we distinguish between coin and banknote distribution, as vehicle capacities only matter when considering the former. Finally, we report on the effectiveness of the valid inequalities in closing the optimality gap at the root node for both the synthetic and the real-life instances and conclude with a sensitivity analysis on the most significant input parameters of our model.

1. Introduction

Routing problems with profits have recently attracted scientific attention in the field of distribution logistics. This shift of interest from classical vehicle routing variations to their profit-oriented counterparts is closely connected to the challenges imposed by real-life settings. Companies currently provide a portfolio of distinct service packages to meet the special needs and preferences of their end-consumers. This, in turn, can make them more competitive by maintaining and growing their end-consumer base, provided operating costs stay sufficiently low. As an immediate consequence of this shift, the revenue obtained by a Logistics Service Provider (LSP) serving a company's (or customer's) service points cannot be considered to be merely analogous to the number of times a service was provided. Rather, it is heavily based on the manpower (associated with the volume) and the skillset (associated with the type of service) needed to provide a delivery to a desired location. Another reason behind this scientific shift lies in the fact that profit-oriented problems consider fleets of limited size. Most supply chains are characterized, to some extent, by demand uncertainties as a result of locality or seasonality effects. Taking into account the limited distribution capacity of any LSP, this can result in poor service levels (for both the LSP and the end-consumers) on days with peak demand, especially when no advanced and problem-specific routing strategies are utilized.

In supply chain management, the term *service level* is typically used to quantitatively assess the performance of a specific element in a supply chain. To come up with such a measure, different Key Performance Indicators (KPIs) are often used in practice. Prominent applications of different KPIs can be found in the agreements between governments and private bus operators. For example,

☆ This manuscript was processed by Associate Editor Pesch.

* Corresponding author.
E-mail addresses: c.orlis@vu.nl (C. Orlis), demetrio.lagana@unical.it (D. Laganá), w.e.h.dullaert@vu.nl (W. Dullaert), daniele.vigo@unibo.it (D. Vigo).

https://doi.org/10.1016/j.omega.2019.02.003
0305-0483/

service level considerations are taken into account in the bus sector between the Australian government and the private bus operators in Sydney to ensure that operators deliver to the market the best possible service levels consistent with stakeholder needs, and especially with the objectives of government [26]. A recent work associating the trade-off between the level of service in the bus network and the incurred operating costs (that are related to the fleet size) in Monterrey in terms of Pareto fronts can be found in [27], while a study dealing with a supplier selection and scheduling problem simultaneously optimizing expecting costs and service levels of customers under disruption risks can be found in [38]. The literature dealing with service level considerations and vehicle routing is scarce with the notable exception of [15] which uses, however, a different model as will be discussed in Section 2. For a recent review measuring, among others, the logistics performance in distributing internal hospital supplies while determining desired service level requirements see [34]. For an extended and more general overview of KPIs used when considering the performance of an LSP see [31]. Generally, Service Level Requirements (SLRs) target either the quality of service offered by a supplier or the quality of service offered to end-consumers. In this study, and motivated by a real-life case, we consider the SLR, imposed by a customer on an LSP to be the minimum-accepted percentage of fulfilled requests over their total number. When this SLR is not met, a predefined financial penalty applies.

Our research was motivated by a real-life cash distribution problem in the Netherlands. Specifically, Cash-In-Transit (CIT) companies face the problem of selecting which customers' service points to visit in order to maximize their profits. Their decision, however, is heavily affected by the SLRs agreed with their customer base (e.g. banks, supermarket chains, retailers, casinos, etc.) and the respective financial penalties in the event that they are missed. The rest of the paper is organized as follows. In Section 2 a literature review is presented with a special focus on vertex-oriented routing problems with profits. Section 3 provides a formal presentation of the Capacitated Routing Problem with Profits and Service Level Requirements (CRPPSLR). In Section 4 the adapted valid inequalities for the polyhedron associated with our problem formulation are presented, along with the branch-and-cut algorithm, including branching and node selections rules. Section 5 contains an extended computational study of a total of 180 synthetic and seven real-life instances of coin and banknote distribution. Finally, we give our concluding remarks in Section 6.

2. Literature review

Given a fleet of vehicles, the main task in the classical setting of the Capacitated Vehicle Routing Problem (CVRP) is to design minimum-cost routes that satisfy the requests of a given set of customers (for an extended overview, see [43]). The problem is known to be $\mathcal{NP}$-complete, as it generalizes simultaneously both the Traveling Salesman Problem (TSP) and the Bin Packing Problem (BPP). In this paper, we consider a problem that generalizes the CVRP and falls into the class of Routing Problems with Profits in which the fleet size is always assumed to be of limited size. Problems of this nature are typically classified into three further subclasses: routing problems in which customers are located on the vertices of a graph; arc routing problems where customers are located on the links (i.e., arcs or edges) of a graph; and vertex-arc routing problems in which customers may be located either on the vertices or on the links (or both) of a graph. In this class of problems, and in contrast to the CVRP setting, it is no longer required to serve all of the customers but only the most profitable subset of them. To arrive at such a decision, the LSP makes use of the information provided by the customer requests regarding their geographical location, the profit obtained by serving each of them, and their demand related to the vehicle capacity. Under this classification, the current vertex-oriented routing problem with profits literature concentrates on the following four main categories of problems based on their objective functions and constraints:

- *The Prize-Collecting TSP (PCTSP), the Prize-Collecting VRP (PCVRP), and the Selective TSP (STSP)*: In the PCTSP, the aim is to maximize the difference between two different objectives, namely, the collected revenue obtained and the total transportation costs incurred by the fulfilled customer requests. However, the problem has appeared in the literature under two more variations in which either of the objectives may be listed as a set of constraints. An overall treatment of this problem including exact solution methods, approximation algorithms, and heuristic approaches can be found in [21]. Surprisingly, the problem is defined differently in its first appearance in 1989 [10]. In this original definition, a customer-specific penalty is occurred when the TSP tour does not visit a customer. The PCVRP extends the PCTSP by seeking prize-collecting TSP routes maximizing the difference between collected revenue and total transportation costs under vehicle capacity constraints. For an example of a rich PCVRP arising in the hot rolling production of the iron and steel industry (and a solution approach based on iterated local search and very large neighborhoods) see [41]. A recent variable neighborhood search heuristic selecting adaptively neighborhoods with higher probability of finding new better solutions can be found in [32] for the version of the problem in which vehicle set-up costs are considered. A problem closely related to the PCTSP is the STSP, for which a branch-and-cut algorithm can be found in [22]. In this problem, some customers may be compulsory, and the aim is to construct a tour of maximal profit whose cost does not exceed a given preset constant.
- *The (Capacitated) Profitable Tour Problem ((C)PTP)*: Similarly to the PCTSP, the objective in the PTP is to maximize the difference between the total collected profit and the total transportation costs incurred by serving the most profitable subset of customers. The PTP was introduced in 1995 [19] along with a presentation of two approximation algorithms for the symmetric version of the problem: one constant-factor based on Linear Programming (LP) rounding [14]; and one purely combinatorial with an input-dependent approximation ratio [23]. In the capacitated version of this problem, a fleet of vehicles is given to serve only the most profitable subset of customers. An exact algorithm based on column generation and heuristics based on tabu search and variable neighborhood search are presented in [5], while a most recent warm-start branch-and-price algorithm can be found in [3]. A new solution framework based on exploring standard VRP neighborhoods of exponential size in pseudo-polynomial time was recently introduced in [45]. Recently, a new version of the problem incorporating strictly non-violated service level requirements for groups of customers, along with a branch-and-price and a population-based heuristic, was presented in [15]. In the case of a single vehicle, instances with up to 800 vertices have been solved to optimality in [28] by utilizing a rich set of valid inequalities in a purely branch-and-cut framework.
- *The (Team) Orienteering Problem ((T)OP) and the Capacitated TOP (CTOP)*: Contrary to the CPTP, the objective function in the TOP maximizes only the total collected profit without considering any transportation costs. On the constraints part, capacity constraints are disregarded, while a hard constraint is imposed on the maximum route duration of each uncapacitated vehicle. This problem was initially designated as

the Multiple Tour Maximum Collection Profit Problem by Butt and Cavalier [16], while the name TOP was coined by Chao et al. [17]. Recently, an effective metaheuristic utilizing a Pareto-dominance criterion controlling the similarity between a generated solution and the incumbent was presented in [29]. From the exact solutions point of view, the most competitive algorithm has been recently presented in [30] and is based on a new formulation that incorporates a polynomial number of variables and constraints. An interesting application that includes the TOP as a special case can be found in paratransit systems, in which, buses operating in a public transport route may diverge from their nominal paths to pick-up passengers with limited mobility and drop them off at their destination [20]. The OP, which first appeared in [24], is the single uncapacitated vehicle version of the above-mentioned problems. A survey treating most problem variants can be found in [44], while an extension of this survey with more up-to-date research findings, problem variants, and solution approaches can be found in [25]. In the capacitated version of this problem, vehicle capacities are taken into account. Exact and heuristic solution approaches for this problem can be found in [5] and [42]. Recently, a new exact approach and a hybrid heuristic for the same problem in which split deliveries are allowed was presented in [4].

– *The Undirected Capacitated General Routing Problem with Profits (UCGRPP)*: In this problem, customer requests may appear either on some of the vertices or on some of the edges (or both) of an undirected graph. This is a general routing problem with profits, meaning that the vertex-oriented version is embedded in the structure of the general version. The objective function of this problem maximizes the difference between the total collected profit and the total transportation costs. Note that the vertex-oriented objective of this problem is exactly the same as that of the multi-vehicle CPTP. On the constraints part, the problem is identical to the CTOP. A recently developed branch-and-cut algorithm for this problem can be found in [2].

An in-depth survey of more classes of vehicle routing problems with profits located on the vertices of the graph can be found in [8] while a study concentrating on problems in which profits may be located on the links (for a representative problem motivated by recreational cyclist searching for nice routes of certain maximum lengths see [40]) of the graph can be found in [6].

In this paper, we introduce and study a real-life vertex-oriented routing problem with profits that generalizes three well-known problems (PCTSP, PCVRP, and CPTP) and their special cases, and in which SLRs are imposed by customers responsible, each of them, for possibly multiple service points. Contrary to the problem described in [15] in which the SLR of every customer should be always fulfilled, our SLR is a minimum-accepted portion of a customer's fulfilled requests over their total number. A penalty rule is activated in the event of a missed SLR. The amount of this penalty is customer-specific, with the aim of incentivizing the LSP to provide service, even in cases when this is not beneficial. On top of this, our problem takes into account maximum route duration constraints imposed by working regulations in many real-life settings.

The abovementioned definition of this SLR is derived from cash distribution practice in the Netherlands. In particular, LSPs are allowed to leave a limited number of ATMs unserved due to exceptional demand or capacity conditions. If an SLR is missed, a flat-rate financial penalty is imposed to the LSP.

Our contributions to the literature are several: (*i*) our problem definition extends the class of routing with profits and generalizes well-established problems in this research stream by (1) allowing customers to be responsible for many requests, extending the standard single-customer single-request pattern, and (2) by introducing the notion of a real-life SLR. To the best of our knowledge, this is the first work that deals with the concept of a real-life KPI, even though different KPIs are already used in practice; (*ii*) we provide a binary programming formulation of the CRPPSLR; (*iii*) we develop a branch-and-cut algorithm by utilizing a set of families of valid inequalities; (*iv*) we adapt and extend theoretical and technical findings from the literature that help solving instances of our problem more effectively; (*v*) we provide 180 new synthetic problem instances; (*vi*) we present a real-life coin and banknote distribution case study, with seven instances coming from the cash supply chain in the Netherlands; (*vii*) we show computational results for our synthetic and real-life instances, along with a sensitivity analysis on the most significant input parameters of our model.

3. Problem description and formulation

The CRPPSLR is a single-period problem defined over an undirected complete graph $G = (V, E)$ where $V = \{0, \dots, N\}$ represents the set of vertices. Traditionally, the vertices of such a graph represent either the customers or the depot(s). In our problem, however, a customer is not represented necessarily by a single vertex. Rather, a non-empty set of vertices is used to represent a customer's service point delivery requests. Consequently, vertices $V' = V \setminus \{0\}$ denote the service points of customer set $\mathcal{C}$ and vertex 0 denotes the depot, while V_C is the set of requests associated with customer $C \in \mathcal{C}$. Therefore, $V' = \bigcup_{C \in \mathcal{C}} V_C$. E denotes the set of edges and is split into $E' = \{(i, j) = V' \times V' : i < j\}$ and $E'' = \{(j, 0) = V' \times \{0\}\} \bigcup \{(0, j) = \{0\} \times V'\}$. In our setting, each customer C belonging to the set of customers $\mathcal{C}$ imposes a service level requirement $0 \le \alpha_C \le 1$. This requirement can be seen as the minimum-allowed percentage of fulfilled requests coming from the same customer. The respective financial penalty when this requirement is not met is denoted by $P_C \ge 0$. An infinite-valued financial penalty is used to indicate a hard constraint on the SLR agreed with a customer imposing compulsory requests. We further consider a homogeneous fleet set $\mathcal{K}$ consisting of vehicle indices, such that each vehicle has capacity equal to $Q > 0$. In the following, we shall denote by c_{ij} and t_{ij} the non-negative cost coefficients and travel times associated with each edge $(i, j) \in E$. Triangle inequality is satisfied for both of these. For the sake of simplicity, we assume that $c_{ij} = t_{ij}$ for all $(i, j) \in E$. A nonnegative fixed profit p_i is associated with each customer's service point demand d_i, with $i \in V'$. The profit of each service point delivery request associated with a customer can be collected at most once by one of the $|\mathcal{K}|$ available vehicles. Finally, each vehicle should start at most one time from the depot, collect revenue as long as this is operationally feasible and profitable, and end at the depot, with a maximum route duration of $T_{\max}$.

Our 0-1 programming formulation uses three families of binary variables: z_i^k with $i \in V$ are equal to 1 if and only if vertex i is visited by vehicle $k \in \mathcal{K}$. Binary variables x_{ij}^k are equal to 1 if and only if edge $(i, j) \in E'$ has been traversed by vehicle k. Accordingly, variables x_{0j}^k and x_{j0}^k with $(0, j)$ and $(0, j) \in E''$ indicate a vehicle k traversal from the depot to vertex j in the former case, and from vertex j to the depot in the later. x_{ij}^{*k} are auxiliary variables equal to x_{ij}^k when $(i, j) \in E'$ and equal to $(x_{0j}^k + x_{j0}^k)$ when $(i, j) \in E''$. Finally, λ_C is equal to 1 if and only if the SLR of customer C is not fulfilled.

The CRPPSLR is then formulated as follows:

$$\text{maximize} \quad \sum_{i \in V'} \sum_{k \in \mathcal{K}} p_i z_i^k - \sum_{(i,j) \in E} \sum_{k \in \mathcal{K}} c_{ij} x_{ij}^{*k} - \sum_{C \in \mathcal{C}} \lambda_C P_C \tag{1}$$

$$\text{s.t.} \quad \sum_{i \in V'} d_i z_i^k \leq Q \qquad \forall k \in \mathcal{K} \tag{2}$$

$$\sum_{j \in V', i<j} x_{ij}^{*k} + \sum_{j \in V, j<i} x_{ji}^{*k} = 2z_i^k \qquad \forall i \in V, \ k \in \mathcal{K} \tag{3}$$

$$\sum_{(i,j) \in E(S)} x_{ij}^{*k} \leq \sum_{i \in S} z_i^k - z_u^k \qquad \forall S \subseteq V, \ |S| \geq 2, \ u \in S, \ \forall k \in \mathcal{K} \tag{4}$$

$$\sum_{k \in \mathcal{K}} z_i^k \leq 1 \qquad \forall i \in V \tag{5}$$

$$\sum_{(i,j) \in E} t_{ij} x_{ij}^{*k} \leq T_{max} \qquad \forall k \in \mathcal{K} \tag{6}$$

$$\alpha_C - \sum_{i \in V_C} \sum_{k \in \mathcal{K}} \frac{z_i^k}{|V_C|} \leq \lambda_C \qquad \forall C \in \mathcal{C} \tag{7}$$

$$z_i^k \in \{0, 1\} \qquad \forall i \in V, \ \forall k \in \mathcal{K} \tag{8}$$

$$x_{ij}^{*k} \in \{0, 1\} \qquad \forall (i,j) \in E, \ \forall k \in \mathcal{K} \tag{9}$$

$$\lambda_C \in \{0, 1\} \qquad \forall C \in \mathcal{C} \tag{10}$$

The objective function (1) seeks to maximize the total profit of the LSP. To calculate this value, we sum up the revenue obtained from the fulfilled service point requests in V' and then substract the total transportation costs incurred by these requests and the (possible) financial penalties for not meeting the service level requirements of the customers. Constraints (2) guarantee that the capacity of each vehicle is not exceeded. Constraints (3) and (4) are constraints eliminating subtours not connected with the depot. Constraints (5) limit the fulfillment of a request by using at most one vehicle, while constraints (6) restrict the maximum duration of a vehicle route. Constraints (7) set the binary variable λ_C to 1 when the service level requirement of customer C is not met. Finally, constraints (8)–(10) define the variable domains.

In the following section we describe the four families of valid inequalities (along with their respective separation procedures) that we adapt for our branch-and-cut algorithm along with the separation strategy and the branching and node selections rules of our overall solution method. Our decision to develop a branch-and-cut solution method was based on the fact that real-world cash distribution scenarios typically require a small amount of routes. The reason behind this observation is that individual routes tend to be long, serving many requests as a result of the vehicle capacity for cash compared to the average demand of a request. Moreover, there are many successful applications of branch-and-cut methods to routing problems with profits (e.g. see [2,7], and [28]) motivating the choice for a branch-and-cut framework for our problem. The main component of our algorithm is a cutting-plane procedure that identifies violated inequalities of several classes, some of which consider the specific nature of the CRPPSLR.

4. Valid inequalities

Our formulation is strengthened with four families of valid inequalities for the polyhedron of the convex hull of the integer vectors satisfying (1)–(10). Observe that, since all the decision variables in (1)–(10) are bounded, the convex hull of these vectors is a polytope. Combinations of these inequalities have been effectively used for solving instances of the CVRP and of other vehicle routing problems, with and without profit considerations (for some representative examples see [1,13,33]). Our decision to include these specific families of inequalities was based on their successful incorporation for solving instances of the UCGRPP [2] that is reducible to our problem when profits are associated exclusively with the vertices of the representation graph. In the following, we present the adaptation of these inequalities to our solution framework along with either exact or heuristic procedures for solving the corresponding separation problems.

4.1. Parity inequalities

Parity inequalities (hereafter PI), also known as co-circuit inequalities [11], have proved useful for problems with binary variables that require the parity of vertices. One successful incorporation, among others, can be found in one of the most well-studied relaxations of the Symmetric TSP [35]. In our case, they are defined for each vehicle index and guarantee that for each subset $S \subseteq V$ and edge cut-set $F \subseteq \delta(S)$, if $|F|$ is odd, at least one further edge must be traversed.

With S being a proper subset of V and $\delta(S) = \{(i,j) \in E : i \in S, \ j \in V \backslash S\}$ containing the edges in the cut between S and $V \backslash S$, disaggregate parity inequalities are defined as follows:

$$\sum_{(i,j) \in \delta(S) \backslash F} x_{ij}^{*k} \geq \sum_{(i,j) \in F} x_{ij}^{*k} - |F| + 1 \quad \forall k \in \mathcal{K}, \ \forall F \subseteq \delta(S), \ |F| \text{ odd} \tag{11}$$

For example, consider the case in which the two edges (i, j) and (j', i) incident to vertex i are traversed by vehicle k. In the presence of solutions with fractional values such that $\bar{z}_i^k = 0.75$, $\bar{x}_{ij}^{*k} = 1$ and $\bar{x}_{j'i}^{*k} = 0.5$, constraint (4) is satisfied, while inequality (11) is violated by $F = \{(i, j)\}$.

We use the framework of Aràoz et al. [1] to separate these inequalities heuristically. In the separation procedure, S is considered to be a singleton rather than a subset of F. More specifically, for each vertex $v \in V$ the edge cut-set $F = \{e \in \delta(v) | \bar{x}_e^* \geq 0.5\}$ is computed and then, if $|F|$ is odd, inequalities (11) are checked for possible violations. If $|F|$ is even, the heuristic makes the cardinality of F odd by either removing or adding an edge to F. This is done by selecting two candidate edges, one from F ($\bar{x}_{e1}^* = \min\{\bar{x}_e^* | e \in F\}$), and one from $\delta(v) \backslash F$ ($\bar{x}_{e2}^* = \max\{\bar{x}_e^* | e \in \delta(v) \backslash F\}$). The decision on which edge to move from one set to another is based on the following control check: if $\bar{x}_{e1}^* - 0.5 \leq 0.5 - \bar{x}_{e2}^*$, e^1 is deleted form F; otherwise, e^2 is added to F. Inequalities (11) can be generalized from a single route to all routes. Inequalities (12) are named aggregate parity inequalities since they involve the variables corresponding to all routes. Note that inequalities (12) are not dominated by (11). Inequalities (12) can be separated heuristically as inequalities (11) by simply aggregating over all the vehicle indices.

$$\sum_{(i,j) \in \delta(S) \backslash F} \sum_{k \in \mathcal{K}} x_{ij}^{*k} \geq \sum_{(i,j) \in F} \sum_{k \in \mathcal{K}} x_{ij}^{*k} - |F| + 1 \qquad \forall F \subseteq \delta(S), \ |F| \text{ odd} \tag{12}$$

Disaggregate and aggregate parity inequalities can also be exactly separated with a polynomial-time algorithm similar to that proposed by Padberg and Rao [36] for finding odd cutsets of minimum weight. However, we decided to separate inequalities (12) heuristically due to their minor expected contribution compared to the computational overhead of separating them exactly.

4.2. Fractional capacity inequalities

Fractional or generalized capacity inequalities (hereafter CI) were introduced in 1998 as part of a branch-and-cut algorithm for the CVRP [9]. Since then, they have been widely used in exact solution frameworks for the CVRP and many more variants of the same problem. Recently, they have also proven to be helpful for solving profit-oriented CVRP variants (see, for example, [2]).

For a given set $S \subseteq V'$ such that $|S| \geq 2$, the following fractional capacity inequalities are valid:

$$\sum_{k\in\mathcal{K}} \sum_{(i,j)\in\delta(S)} x_{ij}^{*k} \geq \frac{2}{Q} \sum_{k\in\mathcal{K}} \sum_{v\in S} d_i z_i^k \tag{13}$$

Let $\left(\bar{z}, \bar{x}^*\right)$ be the fractional solution of the linear programming relaxation. The related exact separation procedure consists of solving a maximum flow problem on a new graph $G'\left(\bar{z}, \bar{x}^*\right)$ constructed from the original G by adding a dummy vertex, denoted $N+1$, connected to every vertex i of G, $i = 0, \ldots, N$. The capacity of each edge (i, j) in $G'\left(\bar{z}, \bar{x}^*\right)$ is denoted by b_{ij} and defined as follows:

$$b_{ij} = \begin{cases} \sum_{k\in\mathcal{K}} \bar{x}_{ij}^{*k}, & \text{for all } (i,j) \in E, \\ \frac{2}{Q} \sum_{k\in\mathcal{K}} d_i \bar{z}_i^k, & \text{such that } i = 0, \ldots, N, \text{ and } j = N+1 \end{cases} \tag{14}$$

Let $S \subseteq V'$ be a set of original vertices. The slack of (13) can be obtained by solving a maximum flow problem on $G'(\bar{x}^*, \bar{z})$, and subtracting $P = \frac{2}{Q} \sum_{i\in V} \sum_{k\in\mathcal{K}} d_i \bar{z}_i^k$ from the capacity of the corresponding minimum cut. If this slack is less than a predefined degree of violation (see Section 4.5.1), then a violation of (13) is checked for the set S that is disjoint from the depot and does not contain the dummy vertex $N+1$.

4.3. Max-time inequalities

Max-time or max-length inequalities (hereafter MTI) were introduced in 2011 for the min-max k-vehicles windy rural postman problem [13]. Since then, they have been effectively used in exact solution frameworks for solving arc routing problems. Some representative problem examples include the Team Orienteering Arc Routing Problem (TOARP) [7] and the UCGRPP [2].

Max-time inequalities are defined as follows: let $F \subset V'$ be a subset of vertices, and let $\sigma(F)$ be the optimal value of the Traveling Salesman Problem (TSP) over the complete graph $G(F)$, that is defined by the depot and all the vertices in F. If $\sigma(F) > T_{\max}$, then the following inequalities hold:

1. All the requests in F cannot be visited by only one vehicle. Therefore, every feasible solution of (1)–(10) must satisfy the following inequality:

$$\sum_{(u,v)\in E(F)} x_{uv}^k \leq |E(F)| - 1 \qquad \forall k \in K \tag{15}$$

where $E(F) = \left\{(i,j) \in E' : i \in F, j \in F\right\}$.

2. If all the vertices inside F are visited ($\sum_{k\in K} \sum_{i\in F} z_i^k = |F|$), then at least two vehicles need to come in F and go out from F. Therefore, every feasible solution of (1)–(10) must satisfy the following inequality:

$$\sum_{k\in K} \sum_{(i,j)\in\delta(F)} x_{ij}^{*k} \geq 4\left(\sum_{k\in K} \sum_{i\in F} z_i^k - |F| + 1\right) \forall F \subseteq V', \quad \text{such that } \sigma(F) > T_{max}, \tag{16}$$

where, for a non-empty subset $F \subseteq V'$, $\delta(F)$ is the non-empty edge cut-set associated with F. More generally, let $n_v(F) = \left\lceil \frac{\sigma(F)}{T_{\max}} \right\rceil$ be the minimum number of vehicles needed to serve all vertices in F, where $\sigma(F) > T_{\max}$, the general form of (16) is:

$$\sum_{k\in K} \sum_{(i,j)\in\delta(F)} x_{ij}^{*k} \geq 2n_v(F)\left(\sum_{k\in K} \sum_{i\in F} z_i^k - |F| + 1\right) \forall F \subseteq V', \quad \text{such that } \sigma(F) > T_{max} \tag{17}$$

Max–time inequalities are worth considering only when $n_v(F) > 1$ for a set of vertices F, and computing $n_v(F)$ involves solving a TSP which is known to be $\mathcal{NP}$-complete. However, despite the inherent computational difficulty of solving TSP instances and the number of times this task needs to be accomplished, it is always highly possible that optimal solutions can be obtained within reasonable computation times by building TSP models and solving them on-the-fly. Clearly, this is due to the fact that most of the TSP instances generated during the separation procedure are of reasonable magnitude. In the case when a subset F is found for which a corresponding inequality (15) is violated, a further check of whether an aggregate max-length inequality (17) is violated for the same set of vertices F follows.

4.4. Symmetry breaking inequalities

Each feasible solution of our problem can be reshuffled on the vehicle indices of the x variables and produce other $|\mathcal{K}|! - 1$ solutions with exactly the same objective value. For example, given a feasible solution, an instance with just seven vehicles can produce $7! = 5040$ equivalent solutions that can be obtained by re-indexing the vehicle indices of the x variables. To break all these symmetries that typically slow down the branch-and-bound search process [39], we introduce the symmetry breaking inequalities (used also in [12])

$$\sum_{(i,j)\in E} c_{ij} x_{ij}^{*k} \geq \sum_{(i,j)\in E} c_{ij} x_{ij}^{*k+1} \qquad \forall k \in \mathcal{K} \backslash \{|\mathcal{K}|\} \tag{18}$$

These inequalities do not allow the re-occurrence of symmetric solution structures by forcing the x_{ij}^{*k} variables with smaller k indices to obtain higher values first. In other words, vehicle $k+1$ is allowed to be dispatched if and only if vehicle k is already in use. Since there are only $|\mathcal{K}| - 1$ such inequalities, we decided to add them all in the initial LP formulation at the root node of the branch-and-cut tree.

4.5. The overall algorithm

4.5.1. The algorithm and separation strategy

To design a computationally efficient branch-and-cut algorithm, it is important to select which inequalities are checked for violation and exactly when this occurs during the search process. After conducting several computational experiments, we came up with the following two observations: (1) adding all the symmetry breaking constraints in the initial LP at the root node does not worsen the overall performance, as there are only $|\mathcal{K}| - 1$ of them; and (2) the root node is a special node when compared to the other nodes of the branch-and-cut tree. More specifically, and based on the computational overhead of each separation routine, we found that separating all the remaining families of valid inequalities at each node of our search tree is not necessarily beneficial. Rather, it seems that separating all of them only at the root node is computationally advantageous. The main reason is that, when the branching takes place, a lot of z_i^k variables are assumed to have integer values and the probability of finding violated capacity inequalities may increase during the branch-and-cut tree exploration. More specifically, at the root node, fractional capacity inequalities (13); parity inequalities (11) and (12); and max-time inequalities (15)–(17) are checked for violation and inserted into our model dynamically. Aggregate parity inequalities (12) are checked for violation only if no disaggregate parity inequalities (11) are found, and max-time inequalities (17) are checked for violation only if there is at least one violated inequality of type (16). All these inequalities help to strengthen the lower bound at the root node of the branch-and-cut tree, and then to reduce branching as much as possible.

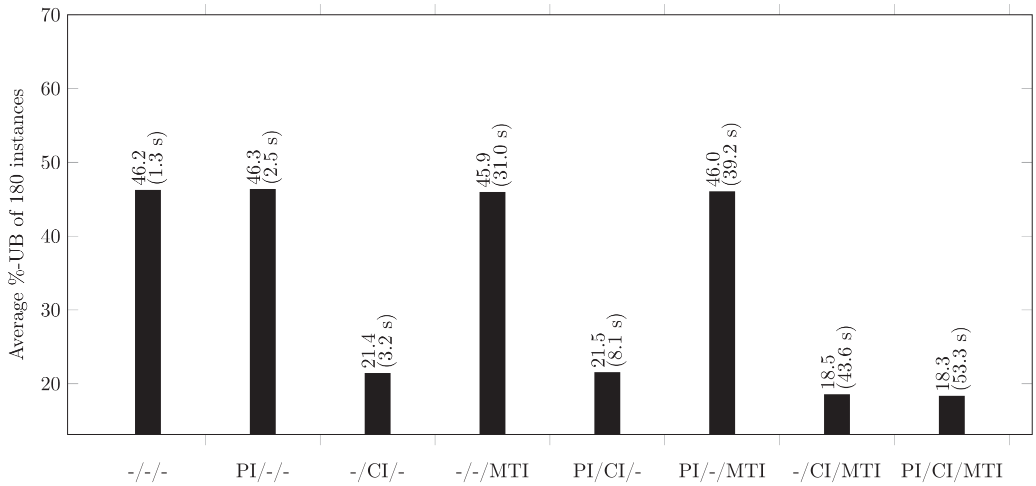

Fig. 1. Effect of families of valid inequalities on closing the optimality gap at the root node for the 180 synthetic instances.

Finally, fractional capacity inequalities (13) are checked for violation at each node of the branch-and-cut tree and added dynamically, together with constraints (4). A limit of 150 total max-time inequalities was imposed. This decision was based on the observation that the improvement produced after a certain number of inequalities was minor compared to the computation time needed to separate them. Finally, subtour elimination constraints and all valid inequalities are considered to be violated only if the slack is less than 0.001. Note that the slack of each inequality represents the positive difference between the two sides of the inequality.

4.5.2. Branching and node selection

For branching we use the *strong branching* strategy implemented in the ILOG CPLEX library. Strong branching is a technique to find the best local variable for branching, and has been proven to work very well in practice. The library allows to assign different priorities to the variables and, at the current iteration, branches on a fractional variable. Variables with higher priority are the first ones checked for branching. For our problem, the priority of variables z_i^k is higher than the priority of variables x_{ij}^{*k}. Each variable z_i^k and x_{ij}^{*k} is further associated with an index that corresponds to its position in the data structure in which it is stored. Therefore, variables z_i^k are ordered in a non-increasing way according to these indices. Then, a decreasing priority is assigned to the elements of this list. A second list is composed by variables x_{ij}^{*k} that are still ordered in the same way, but starting from a priority level that is less than the smallest priority in the first list. Our approach utilizes the *best bound first* node selection strategy in which the selected node is guaranteed to be that with the biggest dual bound. This strategy is known to produce small-sized branch-and-bound trees.

5. Computational experiments

In this section we test our branch-and-cut algorithm and examine the performance of the proposed valid inequalities from Section 4. To this end, we generate 180 synthetic instances derived from [18] and then present a real-life case of coin and banknote distribution from the Dutch cash supply chain. All synthetic instances are derived from seven instances in which a big part of our model's input parameters is provided. For the synthetic instances, we further present a sensitivity analysis associating, among others, the computational difficulty and the objective values obtained with a portion of our model's input parameters.

Our proposed branch-and-cut algorithm was implemented in Java (JDK 8) and tested on a single core of an Intel Core i7-6700U running at 4.00 GHz, equipped with 24 GB of memory. All instances were solved by using CPLEX 12.7, for which all default cuts were activated. A CPU time limit (TL) of two hours was imposed for the branch-and-cut algorithm. The respective gap between the best dual bound and the best solution found until TL is also reported as equal to $\frac{|\overline{\theta}-\theta|}{|\overline{\theta}|} \cdot 100\%$, where $\overline{\theta}$ is the best dual bound produced by CPLEX and θ the best solution found within the TL. The computational results for the synthetic and real-life instances are shown in Figs. 1 and 3 and in Tables 2–4 and 6, where the meaning of column headings is as follows:

Name	Name of the problem instance
opt	best dual bound computed within the TL
pre-UB	initial linear relaxation value of the problem
#r-PI	number of parity inequalities added at root node
#r-CI	number of capacity inequalities added at root node
#r-MTI	number of max-time inequalities added at root node
r-UB	best dual bound obtained at root node
r-time (s)	total computing time (in seconds) spent at root node
%-UB	percentage ratio $\frac{\lvert\text{r-UB}-\text{opt}\rvert}{\lvert\text{opt}\rvert}$
#CI	number of capacity inequalities added after root node
%g-SEC	number of exact subtour elimination constraints added in the branch–and–cut tree
%g-LSEC	number of lazy subtour elimination constraints added in the branch–and–cut tree
nodes	number of nodes in the branch-and-cut tree
%-visited	percentage of fulfilled requests for customers of type *a* and *b* (see Section 5.1)
solved	number of instances solved out of a set of instances
%-gap	percentage gap
time (s)	total computing time (in seconds)

5.1. Synthetic instance generation

To test the computational behavior of our branch-and-cut algorithm we derive 180 instances by making use of five Distance-Constrained CVRP (DCCVRP) instances that can be found in Christofides et al. [18] (available in http://vrp.atd-lab.inf.puc-rio.br/index.php/en/). These instances (namely CMT6, CMT7, CMT8, CMT9, and CMT10) provide information about (1) the number of available vehicles; (2) the capacity of the vehicles; (3) the constraint on the maximum duration for each vehicle; and (4) the Euclidean 2-dimensional coordinates and demands of the requests and of the depot (only coordinates). In each of the five instance sets, comprised of 36 instances, the selected service point requests were chosen randomly. Furthermore, larger instances belonging to the same instance set always contain the service points of the smaller instances. Similarly to the decision of the authors in [3,5],

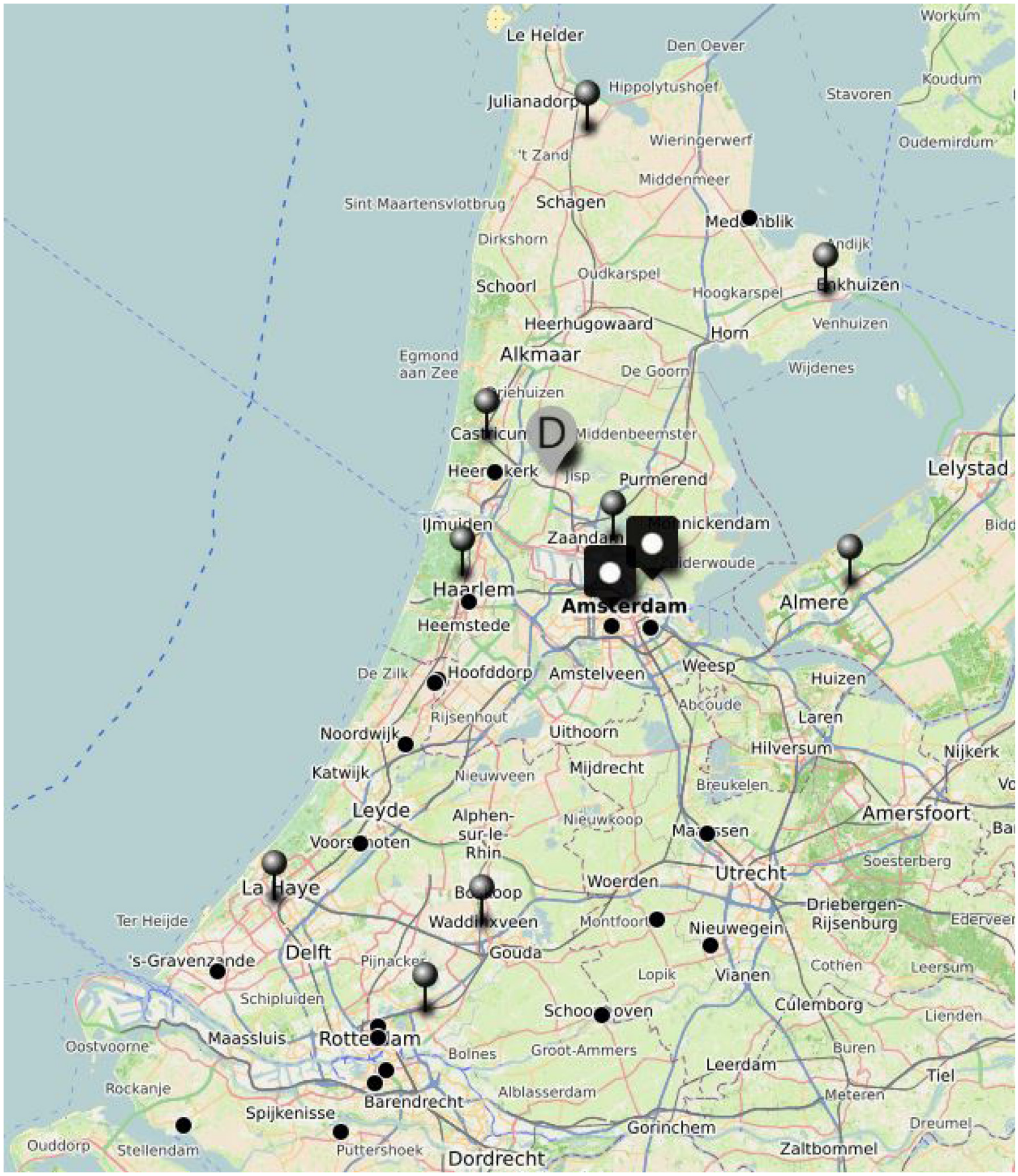

Fig. 2. Coin distribution instance (CDI3) with 21 retailer requests without any SLR (black dots), 9 requests for coin replenishment with an SLR equal to 80% (pinned locations), and 2 obligatory request from a casino and a retailer whose request is soon to reach the maximum-allowed periods without delivery (black rectangles with a white dot in the middle).

Table 1
General structure of the instances given a specific SLR and penalty for the type *b* customer.

	CMT6		CMT7		CMT8		CMT9		CMT10	
$\lvert V'\rvert$	Q=160, $T_{\max}$=200	$\lvert\mathcal{K}\rvert$	Q=140, $T_{\max}$=160	$\lvert\mathcal{K}\rvert$	Q=200, $T_{\max}$=230	$\lvert\mathcal{K}\rvert$	Q=200, $T_{\max}$=200	$\lvert\mathcal{K}\rvert$	Q=200, $T_{\max}$=200	$\lvert\mathcal{K}\rvert$
20	d_1: 14 (a), 5 (b), 1 (c) d_2: 12 (a), 6 (b), 2 (c) d_3: 10 (a), 7 (b), 3 (c)	3	d_1: 14 (a), 5 (b), 1 (c) d_2: 12 (a), 6 (b), 2 (c) d_3: 10 (a), 7 (b), 3 (c)	4	d_1: 14 (a), 5 (b), 1 (c) d_2: 12 (a), 6 (b), 2 (c) d_3: 10 (a), 7 (b), 3 (c)	2	d_1: 14 (a), 5 (b), 1 (c) d_2: 12 (a), 6 (b), 2 (c) d_3: 10 (a), 7 (b), 3 (c)	2	d_1: 14 (a), 5 (b), 1 (c) d_2: 12 (a), 6 (b), 2 (c) d_3: 10 (a), 7 (b), 3 (c)	2
25	d_1: 17 (a), 6 (b), 2 (c) d_2: 15 (a), 7 (b), 3 (c) d_3: 12 (a), 8 (b), 5 (c)	4	d_1: 17 (a), 6 (b), 2 (c) d_2: 15 (a), 7 (b), 3 (c) d_3: 12 (a), 8 (b), 5 (c)	4	d_1: 17 (a), 6 (b), 2 (c) d_2: 15 (a), 7 (b), 3 (c) d_3: 12 (a), 8 (b), 5 (c)	3	d_1: 17 (a), 6 (b), 2 (c) d_2: 15 (a), 7 (b), 3 (c) d_3: 12 (a), 8 (b), 5 (c)	3	d_1: 17 (a), 6 (b), 2 (c) d_2: 15 (a), 7 (b), 3 (c) d_3: 12 (a), 8 (b), 5 (c)	3
30	d_1: 21 (a), 7 (b), 2 (c) d_2: 18 (a), 9 (b), 3 (c) d_3: 15 (a), 10 (b), 5 (c)	4	d_1: 21 (a), 7 (b), 2 (c) d_2: 18 (a), 9 (b), 3 (c) d_3: 15 (a), 10 (b), 5 (c)	5	d_1: 21 (a), 7 (b), 2 (c) d_2: 18 (a), 9 (b), 3 (c) d_3: 15 (a), 10 (b), 5 (c)	3	d_1: 21 (a), 7 (b), 2 (c) d_2: 18 (a), 9 (b), 3 (c) d_3: 15 (a), 10 (b), 5 (c)	3	d_1: 21 (a), 7 (b), 2 (c) d_2: 18 (a), 9 (b), 3 (c) d_3: 15 (a), 10 (b), 5 (c)	3

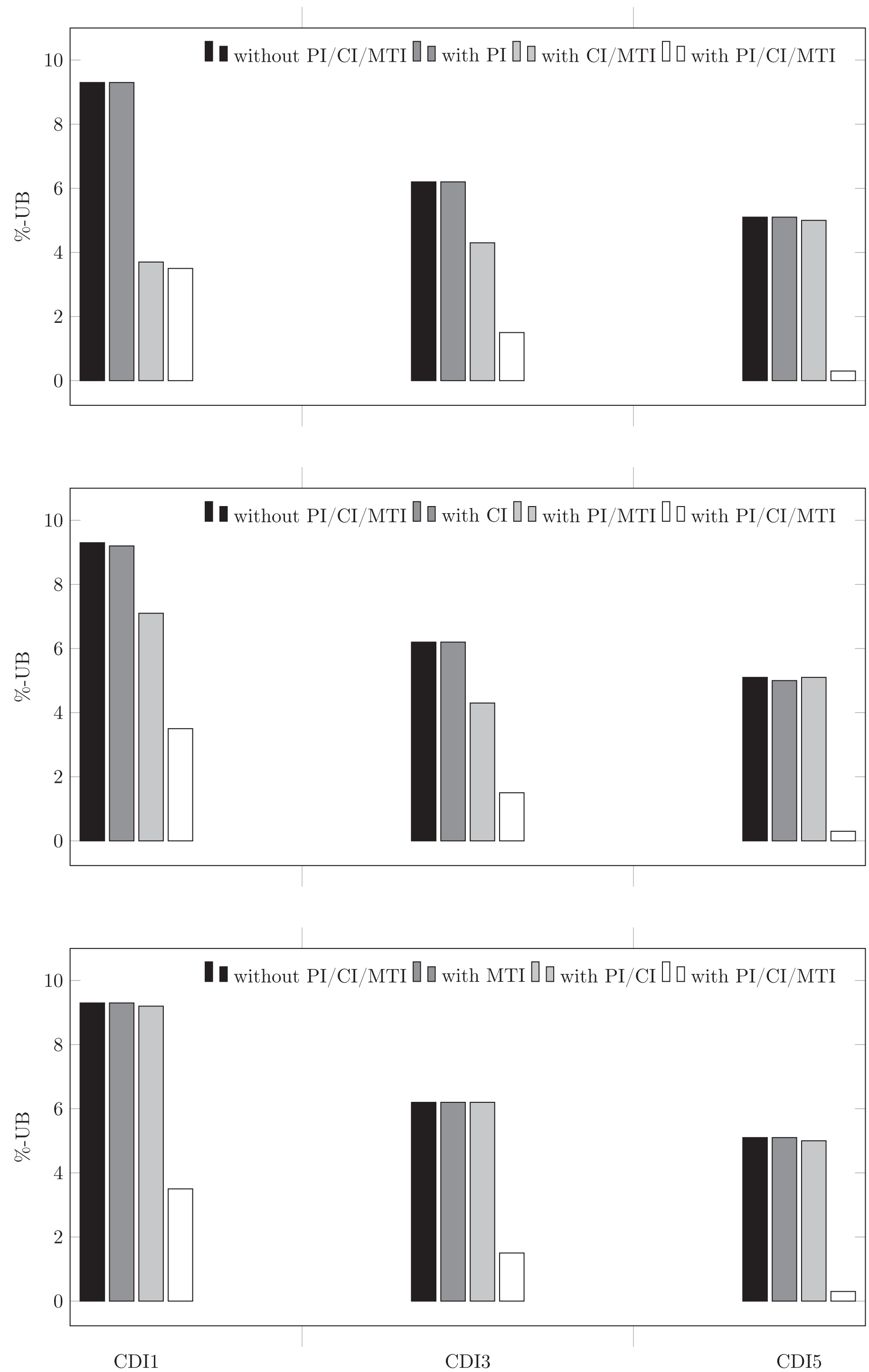

Fig. 3. Effect of valid inequalities on closing the optimality gap at the root node for three coin distribution instances (top: effect of PI, middle: effect of CI, bottom: effect of MTI).

and [15], our decision to utilize this set of DCCVRP instances was based on the fact that most of the input parameters of our model are present in the data set. However, we decided to keep the original number of vehicles and change it only proportionally based on the size of the 36 instances derived from each DCCVRP instance.

Given the novel nature of our problem, we modify these original instances by determining our problem's specific input parameters. We follow the guidelines of [4] and [5] for assigning profits to the service point requests of the customers. Both of the abovementioned papers make use of instances that can be also be found in [18] for generating synthetic instances with suitable structures for the CTOP (and the variant of CTOP in which split deliveries are allowed) and for the CPTP.

We further consider three types of customers (a, b, and c) with different service level requirements. Customers of type a impose no SLR at all, while customers of type c are associated with an SLR equal to 100% and an infinite-valued penalty when not receiving service. Finally, the type b customer is associated with an SLR equal either to 50% ($\alpha_1 = 0.5$) or 80% ($\alpha_2 = 0.8$). Both SLRs for the type b customer are associated with penalty values equal to

20 (p_1) and 150 (p_2) units. For each of the five original instances, 36 were generated by considering all combinations of the following input data: (1) three different instance sizes (i.e., with 20, 25, and 30 service point delivery requests); (2) three different allocation rules (d_1, d_2, and d_3) for the requests between the three customer types; (3) two different SLRs (α_1 and α_2); and (4) 2 different penalty values (p_1 and p_2) for not meeting the SLR of the type b customer. In d_1, requests were distributed by allocating $\lfloor 0.7|V'|\rfloor$ to customers of type a, $\lfloor 0.25|V'|\rfloor$ to customer of type b, and $|V'| - \lfloor 0.7|V'|\rfloor - \lfloor 0.25|V'|\rfloor$ to type c customers. In d_2, $\lfloor 0.6|V'|\rfloor$ requests were assigned to customers of type a, $\lfloor 0.3|V'|\rfloor$ to customer of type b, and $|V'| - \lfloor 0.6|V'|\rfloor - \lfloor 0.3|V'|\rfloor$ to type c customers. Finally, in d_3, $\lfloor 0.5|V'|\rfloor$ requests were assigned to customers of type a, $\lfloor 0.35|V'|\rfloor$ to customer of type b, and $|V'| - \lfloor 0.5|V'|\rfloor - \lfloor 0.35|V'|\rfloor$ to type c customers. The profit of each request was set equal to $p_i = (0.5 + h)d_i$, where h is a random number between 0 and 1 and d_i is the demand of service point $i \in V'$. A general description of the instances regarding the allocation of requests among the different customer types and the instance-specific input parameters is given in Table 1.

5.1.1. Synthetic instance results

Tables 2–4 report on the average computational behavior of our branch-and-cut algorithm for a total set of 180 synthetic instances. The numbers are rounded to one decimal point. These results do not take into account the values of #g-SEC and #g-LSEC for four instances (CMT7-30-$d_2 - \alpha_1 - p_1$, CMT7-30-$d_2 - \alpha_1 - p_2$, CMT7-30-$d_2 - \alpha_2 - p_1$, and CMT7-30-$d_2 - \alpha_2 - p_2$) for which a feasible solution was not found by CPLEX within the TL, even if one does exist for all of them.

The name of each instance provides information about (1) the original instance; (2) the distribution rule for allocating the requests among the three types of customers; (3) the SLR of the type b customer; and (4) the penalty that needs to be incurred by the LSP in case of missing the SLR of the type b customer. For example, CMT6-20-$d_1 - \alpha_1 - p_1$ is an instance produced by CMT6 in which 20 requests were allocated to customers based on the d_1 rule, and in which the LSP incurs a penalty equal to p_1 in case of missing the SLR of the type b customer which is equal to α_1. CMT6->10-20-$d_1 - \alpha_1 - p_1$, accordingly, reports the average computational results of CMT6 up to CMT10 given a fixed number of total service points, the distribution rule, the SLR of the type b customer, and the penalty for missing this specific SLR. For more detailed computational results see Tables A7–A11 in the Appendix A.

Fig. 1 highlights the reason behind our decision to separate the three families of valid inequalities at the root node without excluding any of them. Along the horizontal axis, we report the families of valid inequalities added during the cutting plane phase at the root node of the branch–and–cut algorithm. More precisely, PI stands for parity inequalities (disaggregate and aggregate), CI stands for fractional or generalized capacity inequalities, and MTI stands for max-time inequalities. Along the vertical axis, we report the average gap between the objective values related to the initial linear relaxations and the objective values of the linear relaxations in which all the corresponding violated inequalities are added. The average time needed for the separation routines is also reported next to the average gap for each family. Each average gap and average time spent (in seconds) is computed based on the gaps and times of all 180 synthetic instances.

We observe that the impact produced by separating all families of valid inequalities is better compared to that produced by separating just CI, while the MTI alone do not provide any significant contribution. This is probably due to the fact that when considering these set of instances, the capacity of the vehicles is more critical compared to the maximum route duration limitation, which, in turn, results in a much higher chance of finding violated CI instead of violated MTI. This is also a further reason behind our decision to separate CI dynamically at each node of the branch-and-cut tree. PI also do not seem to produce any result either when used in the absence of CI and MTI. Our previous observation about the effectiveness of the CI are verified when observing the gaps produced by incorporating any two families of valid inequalities. However, it appears that all (PI, CI, and MTI) together manage to close the optimality gap better. This indicates that there are synergistic effects when they are separated in the order described in Section 4.5.1. Finally, it seems that PI and MTI do not produce any significant impact in closing the gap for the synthetic instances. As we will see later, this is different when considering the coin distribution instances of Section 5.2.1. There it seems that PI and MTI together work synergistically and manage to close the optimality gap satisfactorily in four out of five instances.

Table 2 presents our main findings. Each row of this table reports the average value of five different instances (coming from different original instances) given a fixed (1) number of vertices; (2) allocation of the requests among the different customer types; (3) SLR for the type b customer; and (4) penalty for missing the SLR of the type b customer.

Overall, the computational performance of our branch-and-cut algorithm is satisfactory. We manage to solve 135 out of the 180 instances to optimality within the two-hour time limit and with an average gap of 9.6%. This average gap, however, drops to 3.5% and the ratio of optimally solved instances to 123/144 when the 36 instances derived from the CMT7 (DCCVRP instance) are excluded. CMT7 (closely associated with the CVRP instance E-n76-k10) is known to be the most difficult of the instances of Christofides et al. [18]. For more details see the computational results of the E instances in [9] and [33]. Additionally, none of the five DCCVRP instances (CMT6-CMT10) with a minimum of 50 service point requests has been solved to optimality by utilizing a pure branch-and-cut solution framework. Thus, solving the more complicated CRPPSLR instances with 30 service points can be considered a challenge. For a more complete picture of branch-and-cut solution framework capabilities see [37]. More specifically, we are able to solve 35 out of the 36 instances of families CMT8 and CMT10, 29 out of 36 instances of family CMT9, 24 out of 36 instances of family CMT6, and 12 out of 36 instances of family CMT7 (for a more detailed presentation of these results see Appendix A).

According to Table 3, SLRs (and their associated penalties) seem to play a very important role in defining the difficulty of the problem, as in most cases the SLR of the type b customer is fulfilled only because of the associated penalty for missing it. We observe that when the SLR of the type b customer is equal to 50% it is fulfilled with an average excess of 18.7%. On the other hand, when the same SLR is equal to 80% it is fulfilled with an average excess of only 0.1%. This indicates that, for this set of 180 problem instances, the LSP would be on average indifferent in an SLR between 0.0% and 68.7%. Additionally, different SLRs for the type b customer result clearly in different profitability levels for the LSP, as indicated by the difference in the “opt” and “%-gap” columns. Furthermore, more demanding SLRs can increase the difficulty of the problem, as this is mainly implied by the different values in the “%-UB”, “nodes”, “%-gap”, and “time” columns. All in all, it seems that the difficulty of the problem generally increases as the SLR of the type b customer increases and the number of service points served does not necessarily increase proportionally to the given SLR for the type b customer.

Table 4 reports on the effect of the two different penalty values for missing the SLR of the type b customer. We observe that there are cases in which it is more beneficial for the LSP to fulfill only the most beneficial requests of the type b customer without meeting the SLR. This is implied by the different values of the “opt” and “%-gap” columns for the two different penalty values. On

Table 2
Summary of the computational results for the 180 instances.

Name	Opt	Pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time	%-UB	#CI	#g-SEC	#g-LSEC	nodes	%-visited	%-gap	solved	Time
CMT6->10-20-$d_1 - \alpha_1 - p_1$	65.9	146.0	196.0	22.4	37.0	80.4	18.6	25.1	117.2	3231.6	12.2	4369.8	(47.0,72.0)	0.0	5/5	89.0
CMT6->10-20-$d_1 - \alpha_1 - p_2$	65.9	146.0	189.8	22.4	37.8	80.4	18.3	25.1	142.4	4164.0	12.4	5545.0	(54.3,76.0)	0.0	5/5	127.7
CMT6->10-20-$d_1 - \alpha_2 - p_1$	62.5	141.5	223.2	16.8	39.4	77.6	17.5	33.2	116.8	2147.4	7.0	3601.0	(52.8,80.0)	0.0	5/5	61.2
CMT6->10-20-$d_1 - \alpha_2 - p_2$	62.5	141.1	213.2	16.0	39.4	78.8	15.1	40.4	156.0	2066.4	7.8	3157.4	(48.9,76.0)	0.0	5/5	48.2
CMT6->10-20-$d_2 - \alpha_1 - p_1$	64.3	141.3	240.0	35.0	49.0	77.4	19.6	22.8	86.4	1362.4	10.4	1777.6	(51.7,65.3)	0.0	5/5	48.9
CMT6->10-20-$d_2 - \alpha_1 - p_2$	64.3	141.3	238.6	35.0	49.0	77.5	19.5	22.9	227.0	3953.4	11.0	4495.0	(51.7,63.3)	0.0	5/5	186.1
CMT6->10-20-$d_2 - \alpha_2 - p_1$	57.2	140.6	249.0	21.8	53.8	74.0	19.3	43.2	202.6	2811.0	8.6	3741.0	(50,8.66.0)	0.0	5/5	90.8
CMT6->10-20-$d_2 - \alpha_2 - p_2$	57.2	140.0	234.0	17.0	45.0	73.2	17.1	41.1	177.8	2611.4	7.8	3468.8	(53.3,83.3)	0.0	5/5	86.7
CMT6->10-20-$d_3 - \alpha_1 - p_1$	58.0	131.9	221.2	19.2	45.8	74.5	17.2	39.5	190.6	3271.4	6.8	4464.4	(46,7.54.0)	0.0	5/5	100.1
CMT6->10-20-$d_3 - \alpha_1 - p_2$	58.0	128.9	214.6	19.4	41.4	75.4	15.9	43.4	216.4	3511.2	5.0	5366.8	(46,7.54.0)	0.0	5/5	107.4
CMT6->10-20-$d_3 - \alpha_2 - p_1$	52.1	149.6	160.8	16.2	41.0	69.2	16.9	55.0	232.4	3140.0	6.8	3776.6	(50.0,85.7)	0.0	5/5	88.9
CMT6->10-20-$d_3 - \alpha_2 - p_2$	52.1	148.6	184.6	16.8	38.8	69.7	15.6	54.8	148.4	2338.0	6.0	2785.4	(52.0,82.8)	0.0	5/5	59.0
CMT6->10-25-$d_1 - \alpha_1 - p_1$	103.3	202.3	376.4	33.2	56.4	123.5	41.7	23.7	441.6	17283.6	13.6	18607.6	(60.0,76.7)	4.2	4/5	1587.4
CMT6->10-25-$d_1 - \alpha_1 - p_2$	103.4	202.3	376.4	33.2	56.4	123.5	42.0	23.7	414.0	16467.0	14.6	17949.2	(56.4,70.0)	8.4	4/5	1550.0
CMT6->10-25-$d_1 - \alpha_2 - p_1$	100.7	202.4	389.8	27.8	73.6	118.2	49.8	20.6	544.2	19592.4	16.2	20553.0	(64.7,86.6)	4.7	4/5	1734.6
CMT6->10-25-$d_1 - \alpha_2 - p_2$	101.1	202.4	360.2	27.2	68.6	118.2	45.9	20.3	412.8	17433.0	12.2	17621.6	(62.3,80.0)	6.7	4/5	1587.9
CMT6->10-25-$d_2 - \alpha_1 - p_1$	100.2	195.6	303.0	27.2	56.2	120.5	41.3	27.9	656.2	23029.0	15.6	26556.4	(54.66,71.4)	11.0	4/5	1810.4
CMT6->10-25-$d_2 - \alpha_1 - p_2$	100.2	190.1	303.0	27.2	56.2	120.5	41.2	27.9	494.0	19684.6	17.4	21786.2	(60.0,68.5)	11.1	4/5	1607.7
CMT6->10-25-$d_2 - \alpha_2 - p_1$	95.2	196.1	473.8	25.2	100.4	114.1	52.4	27.0	513.0	18679.2	8.2	19511.4	(62.7,82.8)	5.8	4/5	1652.4
CMT6->10-25-$d_2 - \alpha_2 - p_2$	95.2	196.0	346.8	31.2	79.2	114.2	59.2	27.0	347.0	16837.4	11.4	18037.0	(62.7,88.6)	3.1	4/5	1568.2
CMT6->10-25-$d_3 - \alpha_1 - p_1$	95.7	193.0	328.0	23.4	53.6	114.2	43.2	27.4	422.8	20572.8	12.8	23445.0	(63.3,62.5)	8.7	4/5	1692.2
CMT6->10-25-$d_3 - \alpha_1 - p_2$	95.7	190.6	328.0	23.4	53.6	114.2	43.0	27.4	465.6	24533.8	11.4	28973.8	(65.0,55.0)	5.7	4/5	1960.6
CMT6->10-25-$d_3 - \alpha_2 - p_1$	87.4	173.7	381.4	34.6	81.8	106.9	46.3	37.0	424.8	18571.8	9.6	20032.0	(60.0,67.5)	9.9	4/5	1586.9
CMT6->10-25-$d_3 - \alpha_2 - p_2$	83.3	173.6	449.6	29.2	116.6	103.7	62.7	147.9	315.8	16923.0	11.4	18771.6	(61.6,85.0)	10.7	4/5	1591.8
CMT6->10-30-$d_1 - \alpha_1 - p_1$	152.0	241.7	962.2	45.6	119.8	171.7	85.5	14.5	1056.8	29777.2	16.8	30798.2	(66.7,68.5)	9.4	2/5	4529.6
CMT6->10-30-$d_1 - \alpha_1 - p_2$	151.7	241.7	962.2	45.6	119.8	171.7	85.9	14.8	1124.2	29355.0	17.8	29218.2	(59.0,65.7)	16.3	2/5	4566.8
CMT6->10-30-$d_1 - \alpha_2 - p_1$	148.3	240.7	1002.0	51.2	125.2	168.6	92.1	15.0	1160.8	34647.2	18.0	36818.2	(59.1,71.4)	26.9	1/5	5795.7
CMT6->10-30-$d_1 - \alpha_2 - p_2$	148.4	246.9	1010.8	51.6	127.8	168.8	102.9	14.9	964.2	31358.0	11.8	33636.2	(55.2,77.1)	21.5	2/5	5021.2
CMT6->10-30-$d_2 - \alpha_1 - p_1$	151.1	248.2	719.2	55.4	111.4	171.0	79.7	13.8	1151.8	28795.8	9.8	31206.0	(61.1,69.4)	15.4	2/5	4937.0
CMT6->10-30-$d_2 - \alpha_1 - p_2$	150.0	246.5	719.2	55.4	111.4	171.0	79.5	15.1	1001.2	28661.8	12.6	31314.6	(58.4,77.8)	8.1	2/5	4848.4
CMT6->10-30-$d_2 - \alpha_2 - p_1$	143.9	244.5	744.2	44.2	122.2	162.7	95.2	14.5	945.2	28966.2	14.0	29421.2	(61.1,80.6)	14.4	2/5	4676.4
CMT6->10-30-$d_2 - \alpha_2 - p_2$	136.5	245.8	699.2	38.6	125.6	158.8	95.0	20.4	948.4	30849.8	18.2	39051.4	(62.5,83.4)	4.8	3/5	4793.5
CMT6->10-30-$d_3 - \alpha_1 - p_1$	145.0	246.4	989.2	53.8	127.4	164.4	107.9	14.6	715.0	22254.4	18.0	23512.2	(68.0,68.0)	13.9	3/5	3509.3
CMT6->10-30-$d_3 - \alpha_1 - p_2$	145.0	244.4	989.2	53.8	127.4	164.4	108.2	14.6	822.0	25911.2	22.6	27322.8	(62.7,70.0)	9.0	3/5	3894.4
CMT6->10-30-$d_3 - \alpha_2 - p_1$	140.6	243.5	733.6	48.0	122.8	163.2	84.5	18.5	854.2	31240.0	14.6	39451.8	(64.0,78.0)	16.9	3/5	4459.4
CMT6->10-30-$d_3 - \alpha_2 - p_2$	138.8	243.8	835.2	53.4	146.0	162.3	122.9	20.8	1058.8	38580.0	10.6	44285.0	(65.3,76.0)	18.0	2/5	6004.0
Average								**18.3**						**9.6**		**2168.3**
#Optima														**135/180**		

Table 3
Aggregated computational results based on the two different service level requirements.

Name	Opt	Pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time	%-UB	#CI	Nodes	%-visited	%-gap	Time
Average (α_1)	99.6	176.3	464.9	34.0	69.4	117.4	47.1	14.4	526.0	18262.9	(57.3,68.7)	6.4	2064.1
#optima ($\boldsymbol{\alpha_1}$)												68/90	
Average (α_2)	93.9	174.9	466.6	30.6	82.6	112.4	52.7	22.3	513.4	19409.7	(58.2,80.1)	12.8	2272.6
#optima ($\boldsymbol{\alpha_2}$)												67/90	

Table 4
Aggregated computational results based on the two different penalties.

Name	Opt	Pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time	%-UB	#CI	Nodes	%-visited	%-gap	Time
Average (p_1)	97.1	175.7	466.8	32.4	75.4	115.0	48.3	16.2	530.7	18980.2	(57.9,74.0)	9.5	2136.1
#optima ($\boldsymbol{p_1}$)												67/90	
Average (p_2)	96.4	175.5	464.7	32.2	76.7	114.7	51.6	20.4	508.7	19146.7	(57.8,74.2)	9.7	2200.5
#optima ($\boldsymbol{p_2}$)												68/90	

Table 5
General description of the seven real-life instances.

Name	$\|V'\|$	$\|\mathcal{K}\|$	Q (coins)	$T_{\max}$ (h)	Distribution	SLR$_b$	P_b
CDI1	26	3	250.000	8	17 (a), 7 (b), 2 (c)	80%	500
CDI2	32	2	250.000	8	21 (a), 9 (b), 2 (c)	80%	500
CDI3	32	4	250.000	8	21 (a), 9 (b), 2 (c)	80%	500
CDI4	35	2	250.000	8	26 (a), 7 (b), 2 (c)	80%	500
CDI5	35	4	250.000	8	26 (a), 7 (b), 2 (c)	80%	500
BDI1	29	3	-	8	19 (a), 7 (b), 3 (c)	80%	700
BDI2	32	2	-	8	25 (a), 5 (b), 2 (c)	80%	700

top of this, the highest penalty value seems to make the problem slightly more difficult as this is indicated by the different values of columns “%-UB” and “%-gap”.

5.2. Case study instances

To assess the behavior of our branch-and-cut algorithm and the possible advantageous or disadvantageous structure of real-life problems, we conducted a case study of the Dutch cash supply chain. The case study is separated into two parts; coin and banknote distribution. The main difference between these two parts is that vehicle capacities must be considered only in coin distribution problems. For banknote distribution, there is a maximum amount of banknotes that are officially allowed to be delivered. However, this amount is never reached, due to the maximum route duration constraints imposed by the working regulations. On top of this, coin and banknote distributions require different types of vehicles.

Banks are among the most important customers of the CIT companies. This is due to (1) the number of requests for coin or Automated Teller Machine (ATM) cash replenishment a CIT receives on a daily basis; and (2) the high average revenue for replenishing coin or ATM devices. CITs, however, are responsible for the delivery of coins and banknotes to several other types of customers, including, but not limited to retailers, casinos, and foreign exchange markets. Typically, requests for coin or ATM cash replenishment that come from banks do impose an SLR slightly below 100% while retailer requests do not impose any SLR. When a retailer request is not fulfilled on the desired day, it typically moves to the next working day (or period). The maximum-allowed number of consecutive working days a retailer request can be missed is described in the agreement between the retailer and the CIT. Our model addresses this issue arising in the Dutch cash supply chain by turning these delayed requests into obligatory requests just prior to their deadline. Finally, there is a third category of customers imposing, among others, obligatory requests that they should never be missed. These customer types typically include banks requesting ATM cash replenishment for devices located in airports, casino companies, and foreign exchange markets.

A general description of the seven real-life coin and banknote distribution instances (CDI and BDI, respectively) can be found in Table 5. Fig. 2 depicts a coin distribution instance (CDI3) with a total of 32 requests of which: 21 come from 4 retailers without SLRs at all; 9 are bank requests for coin replenishment of dedicated devices with an SLR of 80%; 1 comes from a customer with an SLR of 100% and an infinite-valued financial penalty for missing it; and 1 is a delayed retailer request that is soon to reach the maximum-allowed periods without delivery. In this set of real-life instances, the type b customer is considered to be the entity representing a number of banks, which, through a joint venture[1], send requests to the CITs for ATM cash replenishment.

5.2.1. Case study results

Fig. 3 highlights the effect of the families of valid inequalities in closing the optimality gap at the root node for the three most difficult coin distribution instances. To do so, a representation exploring all possible combinations of families of valid inequalities is considered.

Our main observation is that no family of valid inequalities is able to close the optimality gap at the root node effectively when used in the absence of the other two. Similarly, the same holds when we consider the optimality gaps produced by using any two families of valid inequalities. For example, CI and MTI are very effective when used together in CDI1 and CDI3 but not in CDI5. The same holds when considering using PI and MTI for the same set of instances. This implies that synergistic effects play an important role in this set of instances as well. No significant contribution is observed by incorporating either PI or MTI (or both applied together) in the banknote distribution instances. Hence, we decided to exclude them from this analysis.

Table 6 reports the performance of our branch-and-cut algorithm for a set of five coin and two banknote distribution instances. Numbers are again rounded to one decimal place. For the coin distribution instances, our branch-and-cut algorithm has proved able to solve instances with up to 35 service point re-

[1] Geldmaat has this role in the scope of the Netherlands.

Table 6
Computational results for the seven real-life instances.

Name	Opt	Pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time (s)	%-UB	#CI	#g-SEC	#g-LSEC	nodes	%-visited	%-gap	Time (s)
CDI1	3623.5	3960.0	1135	41	150	3748.6	132.6	3.5	973	72,538	17	71,634	(88.2,85.7)	0.5	TL
CDI2	4412.9	4414.6	6364	71	0	4413.2	1781.9	0.0	492	65,923	27	64,243	(100.0,100.0)	0.0	TL
CDI3	4156.2	4414.6	8841	336	150	4219.1	2046.2	1.5	138	21,964	31	21,317	(85.7,88.9)	2.1	TL
CDI4	5213.7	5283.8	2398	42	0	5261.7	985.3	0.0	689	38,459	40	37,418	(92.3,100.0)	0.0	TL
CDI5	5131.2	5392.0	5465	94	150	5146.8	752.2	0.3	204	24,148	28	23,788	(76.9,71.4)	42.4	TL
BDI1	2891.7	2898.2	75	–	7	2891.8	6.9	0.0	–	30,477	11,636	56,745	(94.7,85.7)	6.4	TL
BDI2	3083.4	3162.9	51	–	8	3083.7	42.2	0.0	–	25,285	1268	61,460	(92.0,100.0)	3.6	TL

quests, with a relatively small gap. For CDI1, CDI2, and CDI4 our code was able to find a solution with a maximum gap of 0.5%, while for CDI3, a solution with a gap of 2.1% was produced within the time limit of two hours. Finally, it is no surprise that we are not able to solve CDI5 with a reasonably small gap, as 35 service point requests seems beyond the capabilities of our code, as indicated in the results of Section 5.1.1) for the synthetic instances with three or more vehicles. This computational limit, however, does not have any practical consequences as real-life instances are normally smaller in size. This is in part due to the fact that the country is split into geographical regions in which up to four vehicles are responsible for the coin distribution tasks. On average, CITs in the Netherlands face coin distribution scenarios with an minimum of 20 and a maximum 32 coin delivery requests per day (CDI4 and CDI5 were artificially created by utilizing CDI2 and CDI3 to show the limitations of our solution framework). Both banknote distribution instances were solved with relatively small gaps. More specifically, BDI1 and BDI2 were solved with gaps equal to 6.4% and 3.6%, respectively. We observe that the difficulty in solving CDI1, CDI3, BDI1, and BDI2 to optimality comes mainly from the fact that we are not able to compute better primal bounds. On the other hand, we observe that strong dual bounds are computed in relatively short times. Overall, computational experiments on the real-life instances confirm the algorithm's performance for the synthetic problem instances.

6. Conclusion

In this paper, we introduced and studied a new routing problem with profits, namely the Capacitated Routing Problem with Profits and Service Level Requirements (CRPPSLR). We proposed a binary programming formulation and described a branch-and-cut algorithm for solving it by utilizing and adapting sets of valid inequalities derived from the literature. A computational study showed the effectiveness of our algorithm through extensive experimentation on 180 synthetic and seven real-life instances.

Our contributions to the literature are several. In terms of problems, and to the best of our knowledge, this was the first study incorporating a real-life Key Performance Indicator (KPI) in the Vehicle Routing Problem-related literature, extending well-known problems such as the Prize-Collecting Traveling Salesman Problem (PCTSP), the Prize-Collecting Vehicle Routing Problem (PCVRP), and the Capacitated Profitable Tour Problem (CPTP). On top of this, we extended the single-vertex single-customer pattern by considering single customers represented by (possibly) many vertices. From a methodological point of view, we developed a branch-and-cut solution framework by incorporating families of valid inequalities with proven effectiveness for problems with similar polyhedral structures; we also provided detailed explanations of their exact and heuristic separation procedures. A set of 180 synthetic instances with up to 30 service points was derived from the well-know instances of Christofides et al. [18]. Computational testing showed the benefits of separating parity, capacity, and max-time inequalities at the root node, and only capacity inequalities thereafter. Furthermore, the impact of model input parameters on the difficulty and profitability of the problem was examined.

Our branch-and-cut algorithm was able to solve 135 out of the 180 synthetic instances within two-hour time limit and with a total average gap of 9.6%. This gap, however, drops to 3.5% and the portion of optimally solved instances rises to 123/144 when the family of 36 instances derived from instance CMT7 is excluded. This instance has previously proved very difficult to solve in [9] and [33]. To evaluate the performance of the algorithm in practice, a real-life case study of coin and banknote distribution was examined. The results supported our decision to separate all families of valid inequalities at the root node, and only capacity inequalities thereafter. Our algorithm was able to solve six out of the seven instances, with up to 35 service point delivery requests close to optimality. More specifically, the first coin distribution instance with 26 requests was solved with a gap equal to 0.5% while the second, third, and fourth coin distribution instances with 32 and 35 requests were solved with gaps equal to 0.0%, 2.1%, and 0.0% respectively. Both banknote distribution instances with 29 and 32 service points requests were solved with relatively small gaps (3.6% and 6.4%). Finally, it was shown how different service level requirements and penalties affect the difficulty and the profitability of the CRPPSLR. More specifically, we show that higher service level requirements and penalty values lead to worse computational performance and lower profitability levels.

Given the importance of the quality of service considerations in the industry, we think it is worthwhile to devote more research activity to handling service level requirements in distribution routing problems. In particular, research could be directed towards modeling different service level measures and alternative penalty structures. Moreover, deriving new problem formulations and developing exact and heuristic solution approaches appear promising directions for further research aimed at unlocking the potential of routing problems with profits and service level requirements.

Acknowledgement

We would like to thank Geldmaat for providing real-life instances of coin and banknote distribution.

Funding

This work was supported by The Dutch Science Foundation [grant number 407-13-050].

Appendix A. Tables of results

Tables A7–A11 report on the computational behavior of our branch-and-cut algorithm on a total set for 180 synthetic instances. Each table reports computational results for the 36 instances produced by the same original instance, as described in Section 5.1. Moreover, the name of each instance provides information about (1) the original instance; (2) the distribution rule for allocating the requests among the three types of customers; (3) the SLR of the type b customer; and (4) the penalty that needs to be incurred by

Table A7
Computational results for the 36 instances produced by CMT6.

Name	Opt	Pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time (s)	%-UB	#CI	#g-SEC	#g-LSEC	Nodes	%-visited	%-gap	Time (s)
CMT6-20-$d_1 - \alpha_1 - p_1$	93.3	145.8	113	16	12	102.8	10.2	10.2	19	63	16	45	(35.0,80.0)	0.0	3.4
CMT6-20-$d_1 - \alpha_1 - p_2$	93.3	145.8	113	16	12	102.8	10.2	10.2	20	62	16	48	(50.0,80.0)	0.0	3.4
CMT6-20-$d_1 - \alpha_2 - p_1$	93.3	145.8	136	14	6	101.2	8.5	8.5	14	58	10	47	(50.0,80.0)	0.0	3.0
CMT6-20-$d_1 - \alpha_2 - p_2$	93.3	145.8	132	12	9	98.9	6.0	6.0	12	52	12	32	(50.0,80.0)	0.0	6.8
CMT6-20-$d_2 - \alpha_1 - p_1$	93.3	145.8	121	15	14	98.9	6.0	6.0	4	45	8	33	(50.0,66.7)	0.0	8.0
CMT6-20-$d_2 - \alpha_1 - p_2$	93.3	145.8	121	15	14	99.3	6.4	6.4	5	45	8	33	(50.0,66.7)	0.0	7.9
CMT6-20-$d_2 - \alpha_2 - p_1$	83.8	144.8	238	27	56	89.9	7.3	7.3	17	83	6	63	(58.3,83.3)	0.0	14.3
CMT6-20-$d_2 - \alpha_2 - p_2$	83.8	144.8	156	12	23	90.6	8.1	8.1	49	193	7	149	(58.3,83.3)	0.0	10.4
CMT6-20-$d_3 - \alpha_1 - p_1$	93.3	145.8	237	25	31	99.0	6.1	6.1	15	60	12	10	(40.0,71.4)	0.0	11.6
CMT6-20-$d_3 - \alpha_1 - p_2$	93.3	145.8	237	25	31	99.0	6.1	6.1	6	53	12	10	(40.0,71.4)	0.0	11.0
CMT6-20-$d_3 - \alpha_2 - p_1$	83.8	144.8	103	13	14	90.6	8.1	8.1	51	167	7	129	(50.0,85.7)	0.0	7.6
CMT6-20-$d_3 - \alpha_2 - p_2$	83.8	144.8	196	14	27	90.9	8.5	8.5	42	176	7	127	(50.0,85.7)	0.0	12.0
CMT6-25-$d_1 - \alpha_1 - p_1$	111.6	165.6	417	17	23	128.4	15.1	15.1	341	2737	18	3047	(41.1,66.7)	0.0	97.5
CMT6-25-$d_1 - \alpha_1 - p_2$	111.6	165.6	417	17	23	128.4	15.1	15.1	275	1925	21	2806	(41.1,66.7)	0.0	73.6
CMT6-25-$d_1 - \alpha_2 - p_1$	110.7	165.6	189	26	25	121.7	9.9	9.9	222	1184	13	1334	(64.7,83.3)	0.0	54.1
CMT6-25-$d_1 - \alpha_2 - p_2$	110.7	165.6	304	23	50	121.8	10.0	10.0	244	2130	10	2087	(64.4,83.3)	0.0	83.8
CMT6-25-$d_2 - \alpha_1 - p_1$	111.6	165.6	198	18	13	128.4	15.1	15.1	373	3078	10	3957	(33.3,71.4)	0.0	102.7
CMT6-25-$d_2 - \alpha_1 - p_2$	111.6	165.6	198	18	13	128.4	15.1	15.1	316	2227	12	3352	(33.3,71.4)	0.0	77.7
CMT6-25-$d_2 - \alpha_2 - p_1$	110.7	165.6	301	26	55	120.2	8.6	8.6	118	904	6	1471	(60.0,85.7)	0.0	53.6
CMT6-25-$d_2 - \alpha_2 - p_2$	110.7	165.6	284	22	54	121.1	9.4	9.4	130	1039	2	1591	(60.0,85.7)	0.0	55.5
CMT6-25-$d_3 - \alpha_1 - p_1$	111.6	165.6	210	18	22	128.4	15.1	15.1	305	1888	18	2961	(33.3,50.0)	0.0	70.7
CMT6-25-$d_3 - \alpha_1 - p_2$	111.6	165.6	210	18	22	128.4	15.1	15.1	358	2214	15	2726	(33.3,50.0)	0.0	81.4
CMT6-25-$d_3 - \alpha_2 - p_1$	110.7	165.6	184	20	47	121.6	9.8	9.8	181	1161	9	1423	(50.0,87.5)	0.0	54.5
CMT6-25-$d_3 - \alpha_2 - p_2$	110.7	165.6	435	25	131	120.4	8.8	8.8	146	1001	13	1247	(33.3,87.5)	0.0	78.8
CMT6-30-$d_1 - \alpha_1 - p_1$	176.5	249.5	2581	36	150	192.0	8.8	8.8	940	50,268	20	51,009	(76.2,71.4)	7.4	TL
CMT6-30-$d_1 - \alpha_1 - p_2$	176.5	249.5	2581	36	150	192.0	8.8	8.8	1037	50,570	19	51,320	(66.7,85.7)	9.3	TL
CMT6-30-$d_1 - \alpha_2 - p_1$	175.4	249.0	2289	30	150	192.2	9.6	9.6	854	48,932	19	48,052	(61.9,85.7)	20.1	TL
CMT6-30-$d_1 - \alpha_2 - p_2$	175.4	248.0	2289	30	150	192.2	9.6	9.6	844	49,613	10	48,707	(71.4,85.7)	21.3	TL
CMT6-30-$d_2 - \alpha_1 - p_1$	174.2	249.5	1628	85	150	191.0	9.6	9.6	956	48,393	11	47,501	(72.2,77.7)	24.3	TL
CMT6-30-$d_2 - \alpha_1 - p_2$	174.2	249.5	1628	85	150	191.0	9.6	9.6	969	48,847	16	48,049	(55.6,100.0)	14.3	TL
CMT6-30-$d_2 - \alpha_2 - p_1$	173.9	248.0	1187	69	150	189.8	9.1	9.1	697	52,477	4	51,871	(66.7,88.9)	15.5	TL
CMT6-30-$d_2 - \alpha_2 - p_2$	173.9	248.0	1024	44	150	189.6	9.0	9.0	856	49,165	13	48,360	(66.7,88.9)	19.4	TL
CMT6-30-$d_3 - \alpha_1 - p_1$	176.0	249.5	2328	32	150	192.2	9.2	9.2	972	47,893	18	48,774	(60.0,70,0)	9.1	TL
CMT6-30-$d_3 - \alpha_1 - p_2$	176.0	249.5	2328	32	150	192.2	9.2	9.2	873	50,730	16	55,100	(60.0,90.0)	4.9	TL
CMT6-30-$d_3 - \alpha_2 - p_1$	175.4	248.6	1482	82	150	190.9	8.8	8.8	754	53,691	13	57,105	(60.0,90.0)	3.1	TL
CMT6-30-$d_3 - \alpha_2 - p_2$	175.4	248.6	1486	82	150	190.9	8.8	8.8	742	54,084	9	61,877	(60.0,90.0)	3.1	TL
Average								**9.7**						**5.9**	**2427.3**
#Optima														**24/36**	

Table A8
Computational results for the 36 instances produced by CMT7.

Name	opt	pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time (s)	%-UB	#CI	#g-SEC	#g-LSEC	nodes	%-visited	%-gap	time (s)
CMT7-20-$d_1-\alpha_1-p_1$	84.0	197.1	477	29	114	115.2	47.7	37.1	356	14,860	4	20,771	(35.7,80.0)	0.0	414.5
CMT7-20-$d_1-\alpha_1-p_2$	84.0	197.1	477	29	114	115.2	46.7	37.1	463	19,327	4	26,420	(57.1,100.0)	0.0	594.0
CMT7-20-$d_1-\alpha_2-p_1$	81.6	197.1	738	30	150	114.2	57.2	40.0	307	7918	4	13,296	(50.0,80.0)	0.0	254.4
CMT7-20-$d_1-\alpha_2-p_2$	81.6	197.1	738	30	150	114.2	57.7	40.0	159	4636	12	8477	(57.1,80.0)	0.0	149.5
CMT7-20-$d_2-\alpha_1-p_1$	76.0	197.1	683	72	150	107.6	52.8	41.6	236	5832	3	8006	(50.0,83.3)	0.0	193.5
CMT7-20-$d_2-\alpha_1-p_2$	76.0	197.1	682	72	150	107.6	51.9	41.6	936	18,848	2	21,613	(50.0,83.3)	0.0	880.1
CMT7-20-$d_2-\alpha_2-p_1$	76.0	197.1	719	33	150	108.8	63.3	43.2	503	9667	2	11,349	(41.7,66.7)	0.0	349.4
CMT7-20-$d_2-\alpha_2-p_2$	76.0	197.1	719	33	150	108.8	62.9	43.2	548	10,166	2	13,086	(50.0,83.3)	0.0	383.3
CMT7-20-$d_3-\alpha_1-p_1$	76.0	197.1	592	21	150	110.6	56.6	45.5	499	13,315	0	18,434	(40.0,85.7)	0.0	443.3
CMT7-20-$d_3-\alpha_1-p_2$	76.0	197.1	592	21	150	110.6	57.0	45.5	512	13,480	0	20,017	(40.0,85.7)	0.0	463.0
CMT7-20-$d_3-\alpha_2-p_1$	76.0	197.1	395	21	106	110.8	38.5	45.8	636	10,176	2	11,565	(50.0,85.7)	0.0	342.8
CMT7-20-$d_3-\alpha_2-p_2$	76.0	197.1	395	21	106	110.8	39.4	45.8	381	8098	5	9529	(40.0,85.7)	0.0	224.3
CMT7-25-$d_1-\alpha_1-p_1$	140.6	252.5	663	46	150	163.2	97.2	16.1	997	66,892	8	66,383	(70.6,100.0)	21.1	TL
CMT7-25-$d_1-\alpha_1-p_2$	140.7	252.5	663	46	150	163.2	97.4	16.0	1010	67,452	8	66,400	(52.9,66.7)	41.8	TL
CMT7-25-$d_1-\alpha_2-p_1$	142.6	252.5	619	49	150	164.9	125.8	15.6	867	65,507	11	64,925	(70.6,100.0)	23.3	TL
CMT7-25-$d_1-\alpha_2-p_2$	144.8	252.5	619	49	150	164.9	126.3	13.9	829	67,914	7	67,110	(70.6,100.0)	33.4	TL
CMT7-25-$d_2-\alpha_1-p_1$	147.7	252.5	683	50	150	164.8	108.1	11.6	1133	75,130	9	73,991	(60.0,100.0)	55.0	TL
CMT7-25-$d_2-\alpha_1-p_2$	147.7	252.5	683	50	150	164.8	108.7	11.6	1131	75,232	9	74,095	(73.3,85.7)	55.7	TL
CMT7-25-$d_2-\alpha_2-p_1$	143.0	252.5	648	41	150	166.0	95.0	16.1	1266	63,869	2	62,766	(66.7,85.7)	28.9	TL
CMT7-25-$d_2-\alpha_2-p_2$	145.3	252.5	648	41	150	166.0	95.0	14.2	1009	69,019	7	64,806	(53.3,100.0)	15.6	TL
CMT7-25-$d_3-\alpha_1-p_1$	143.3	252.5	781	51	150	161.7	100.1	12.8	745	72,836	7	72,100	(75.0,75.0)	43.5	TL
CMT7-25-$d_3-\alpha_1-p_2$	143.4	252.5	781	51	150	161.7	100.6	12.8	785	74,344	5	73,596	(83.3,50.0)	28.5	TL
CMT7-25-$d_3-\alpha_2-p_1$	135.6	252.4	562	32	150	154.0	117.1	13.6	784	70,395	3	69,600	(66.7,50.0)	49.3	TL
CMT7-25-$d_3-\alpha_2-p_2$	133.4	252.4	565	28	150	152.0	138.7	13.9	598	64,724	4	64,102	(58.3,87.5)	99.6	TL
CMT7-30-$d_1-\alpha_1-p_1$	195.0	324.4	685	42	150	211.5	132.4	8.5	693	30,179	6	29,562	(71.4,85.7)	39.7	TL
CMT7-30-$d_1-\alpha_1-p_2$	193.7	324.4	685	42	150	211.5	132.1	9.2	724	25,214	9	24,569	(42.9,57.1)	49.6	TL
CMT7-30-$d_1-\alpha_2-p_1$	191.2	324.4	745	43	150	206.6	125.7	8.1	595	21,876	6	21,444	(52.4,42.9)	63.0	TL
CMT7-30-$d_1-\alpha_2-p_2$	188.4	324.4	745	43	150	206.6	125.7	9.7	654	19,717	1	19,157	(19.0,42.9)	168.5	TL
CMT7-30-$d_2-\alpha_1-p_1$	191.2	324.4	720	45	150	208.5	149.6	9.0	696	20,042	4	20,004	(-,-)	-	TL
CMT7-30-$d_2-\alpha_1-p_2$	190.5	324.4	720	45	150	208.5	148.6	9.4	692	19,899	5	19,880	(-,-)	-	TL
CMT7-30-$d_2-\alpha_2-p_1$	186.8	323.2	733	46	150	199.1	152.1	6.6	706	20,732	8	20,890	(-,-)	-	TL
CMT7-30-$d_2-\alpha_2-p_2$	181.3	323.2	733	38	150	197.7	155.6	9.0	702	20,334	5	20,844	(-,-)	-	TL
CMT7-30-$d_3-\alpha_1-p_1$	187.4	324.4	720	43	150	208.0	135.5	11.0	763	21,377	6	20,680	(73.3,60.0)	60.5	TL
CMT7-30-$d_3-\alpha_1-p_2$	187.5	324.4	720	43	150	208.0	137.3	10.9	706	21,481	3	20,834	(46.7,50.0)	40.2	TL
CMT7-30-$d_3-\alpha_2-p_1$	186.5	324.4	600	45	150	205.0	149.7	9.9	555	23,897	6	23,224	(46.7,40.0)	81.4	TL
CMT7-30-$d_3-\alpha_2-p_2$	189.5	324.4	606	41	150	205.9	159.3	8.7	501	24,931	1	24,423	(46.6,40.0)	126.2	TL
Average								**17.1**						**45.4**	**4930.3**
#Optima														**12/36**	

Table A9
Computational results for the 36 instances produced by CMT8.

Name	Opt	Pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time (s)	%-UB	#CI	#g-SEC	#g-LSEC	Nodes	%-visited	%-gap	Time (s)
CMT8-20-$d_1-\alpha_1-p_1$	45.6	94.9	153	15	3	53.3	14.7	16.9	10	67	13	56	(50.0,80.0)	0.0	1.5
CMT8-20-$d_1-\alpha_1-p_2$	45.6	94.9	153	15	3	53.3	14.7	16.9	12	69	14	63	(50.0,80.0)	0.0	14.9
CMT8-20-$d_1-\alpha_2-p_1$	45.6	94.9	56	7	0	49.7	1.4	9.0	7	21	10	24	(50.0,80.0)	0.0	1.5
CMT8-20-$d_1-\alpha_2-p_2$	45.6	94.9	51	12	1	48.9	2.1	7.2	12	38	9	47	(28.6,60.0)	0.0	2.3
CMT8-20-$d_2-\alpha_1-p_1$	45.6	94.9	121	27	2	55.4	6.8	21.5	34	171	15	216	(50.0,66.7)	0.0	7.4
CMT8-20-$d_2-\alpha_1-p_2$	45.6	94.9	115	27	2	55.4	6.4	21.5	71	216	20	200	(50.0,66.7)	0.0	7.1
CMT8-20-$d_2-\alpha_2-p_1$	39.9	94.9	83	13	6	45.4	5.0	13.8	35	102	17	88	(50.0,83.3)	0.0	5.3
CMT8-20-$d_2-\alpha_2-p_2$	39.9	94.9	89	11	0	41.3	2.4	3.5	1	28	8	20	(50.0,83.3)	0.0	2.4
CMT8-20-$d_3-\alpha_1-p_1$	29.5	79.5	65	12	0	41.5	4.0	40.7	103	433	7	789	(50.0,57.1)	0.0	7.0
CMT8-20-$d_3-\alpha_1-p_2$	29.5	79.5	71	20	0	41.5	5.8	40.7	101	249	7	233	(50.0,57.1)	0.0	6.9
CMT8-20-$d_3-\alpha_2-p_1$	27.4	79.5	100	16	13	30.6	6.6	11.7	6	32	9	25	(50.0,85.7)	0.0	6.7
CMT8-20-$d_3-\alpha_2-p_2$	27.4	79.5	142	20	10	29.9	8.0	9.1	8	38	8	5	(50.0,71.4)	0.0	8.1
CMT8-25-$d_1-\alpha_1-p_1$	65.8	107.7	221	29	13	82.4	39.9	25.2	139	655	27	735	(47.1,66.7)	0.0	50.3
CMT8-25-$d_1-\alpha_1-p_2$	65.8	107.7	221	29	13	82.4	40.6	25.2	206	946	29	975	(47.1,66.7)	0.0	56.7
CMT8-25-$d_1-\alpha_2-p_1$	58.7	106.6	268	27	28	66.9	46.2	14.0	125	536	31	469	(41.1,83.3)	0.0	55.3
CMT8-25-$d_1-\alpha_2-p_2$	58.7	106.6	211	21	6	67.0	36.1	14.1	111	435	12	400	(41.1,66.7)	0.0	42.5
CMT8-25-$d_2-\alpha_1-p_1$	43.2	92.4	206	37	29	64.7	32.1	49.8	336	2590	26	3709	(46.7,57.1)	0.0	90.3
CMT8-25-$d_2-\alpha_1-p_2$	43.2	92.4	206	37	29	64.7	32.2	49.8	258	1734	34	2703	(46.7,57.1)	0.0	65.2
CMT8-25-$d_2-\alpha_2-p_1$	34.9	91.3	237	19	53	50.2	42.6	43.8	255	2324	6	2585	(33.3,71.4)	0.0	87.9
CMT8-25-$d_2-\alpha_2-p_2$	34.9	91.3	120	20	16	49.7	19.1	42.4	137	926	13	1558	(46.7,85.7)	0.0	35.2
CMT8-25-$d_3-\alpha_1-p_1$	38.6	92.4	218	16	13	54.8	29.4	42.0	174	1400	22	1824	(50.0,50.0)	0.0	53.1
CMT8-25-$d_3-\alpha_1-p_2$	38.6	92.4	218	16	13	54.8	28.5	42.0	195	1673	20	2068	(50.0,50.0)	0.0	59.8
CMT8-25-$d_3-\alpha_2-p_1$	18.6	91.2	217	21	33	36.3	38.0	95.2	242	1943	28	2568	(50.0,50.0)	0.0	77.0
CMT8-25-$d_3-\alpha_2-p_2$	3.6	91.2	519	25	150	26.8	88.3	644.4	273	6041	24	8080	(83.3,87.5)	0.0	274.6
CMT8-30-$d_1-\alpha_1-p_1$	114.8	184.3	212	22	22	131.9	87.1	14.9	454	3559	20	3497	(47.6,71.4)	0.0	217.8
CMT8-30-$d_1-\alpha_1-p_2$	114.8	184.3	212	22	22	131.9	87.0	14.9	455	3756	21	4418	(47.6,71.4)	0.0	225.3
CMT8-30-$d_1-\alpha_2-p_1$	113.2	183.4	341	48	26	126.8	117.1	12.0	302	2075	16	3376	(47.6,85.7)	0.0	178.4
CMT8-30-$d_1-\alpha_2-p_2$	113.2	183.4	253	28	39	127.3	88.1	12.5	190	1865	16	3131	(47.6,85.7)	0.0	142.5
CMT8-30-$d_2-\alpha_1-p_1$	114.8	184.3	298	40	37	129.7	111.1	13.0	422	3036	8	2805	(50.0,55.6)	0.0	210.1
CMT8-30-$d_2-\alpha_1-p_2$	114.8	184.3	298	40	37	129.7	110.4	13.0	310	1841	8	2606	(50.0,55.6)	0.0	167.5
CMT8-30-$d_2-\alpha_2-p_1$	94.8	182.3	331	31	79	109.9	125.7	15.9	619	4086	21	4763	(50.0,55.6)	0.0	286.5
CMT8-30-$d_2-\alpha_2-p_2$	67.4	182.3	440	35	103	93.3	154.8	38.4	1006	34,667	31	49,329	(61.1,88.9)	0.0	4627.8
CMT8-30-$d_3-\alpha_1-p_1$	104.8	183.2	847	106	127	117.8	264.2	12.4	269	3328	28	5813	(60.0,50.0)	0.0	361.7
CMT8-30-$d_3-\alpha_1-p_2$	104.8	183.2	847	106	127	117.8	264.0	12.4	492	6771	57	6671	(60.0,50.0)	0.0	517.0
CMT8-30-$d_3-\alpha_2-p_1$	89.8	182.1	416	45	73	115.8	111.3	29.0	613	21,187	32	29,960	(66.7,80.0)	0.0	1438.9
CMT8-30-$d_3-\alpha_2-p_2$	80.2	182.1	446	46	130	110.4	180.3	37.7	1222	46,910	12	59,666	(66.7,80.0)	2.9	TL
Average								**22.3**						**0.1**	**460.3**
#Optima														**35/36**	

Table A10
Computational results for the 36 instances produced by CMT9.

Name	Opt	Pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time (s)	%-UB	#CI	#g-SEC	#g-LSEC	nodes	%-visited	%-gap	Time (s)
CMT9-20-$d_1 - \alpha_1 - p_1$	32.8	107.9	119	37	42	49.3	17.8	50.3	180	1114	22	950	(57.1,60.0)	0.0	23.2
CMT9-20-$d_1 - \alpha_1 - p_2$	32.8	107.9	119	37	42	49.3	17.8	50.3	184	1284	23	1143	(57.1,60.0)	0.0	23.9
CMT9-20-$d_1 - \alpha_2 - p_1$	24.1	105.8	134	26	34	47.6	18.7	97.5	241	2703	9	4611	(57.1,80.0)	0.0	45.7
CMT9-20-$d_1 - \alpha_2 - p_2$	24.1	105.8	74	17	15	57.6	7.1	139.0	592	5579	2	7206	(51.7,80.0)	0.0	79.5
CMT9-20-$d_2 - \alpha_1 - p_1$	32.8	107.9	198	48	65	44.2	30.2	34.8	141	720	15	610	(58.3,50.0)	0.0	33.1
CMT9-20-$d_2 - \alpha_1 - p_2$	32.8	107.9	198	48	65	44.2	30.4	34.8	106	614	14	606	(58.3,50.0)	0.0	32.8
CMT9-20-$d_2 - \alpha_2 - p_1$	23.1	105.8	141	22	44	55.7	18.6	141.1	441	4157	15	7182	(50.0,83.3)	0.0	82.3
CMT9-20-$d_2 - \alpha_2 - p_2$	23.1	105.8	73	14	14	55.5	6.6	140.3	286	2637	13	4071	(50.0,83.3)	0.0	31.9
CMT9-20-$d_3 - \alpha_1 - p_1$	25.1	106.7	158	29	40	49.2	17.8	96.0	292	2471	13	3052	(60.0,57.1)	0.0	37.0
CMT9-20-$d_3 - \alpha_1 - p_2$	25.1	106.7	119	22	18	54.0	8.9	115.1	435	3711	3	6545	(60.0,57.1)	0.0	54.2
CMT9-20-$d_3 - \alpha_2 - p_1$	16.3	100.0	130	18	51	47.7	29.0	192.6	449	5214	11	7044	(60.0,85.7)	0.0	84.5
CMT9-20-$d_3 - \alpha_2 - p_2$	16.3	95.1	104	17	39	47.0	20.2	188.3	303	3343	8	4242	(70.0,85.7)	0.0	48.3
CMT9-25-$d_1 - \alpha_1 - p_1$	54.4	164.1	311	45	61	81.5	41.6	49.8	597	15,039	7	20,960	(76.4,83.3)	0.0	554.1
CMT9-25-$d_1 - \alpha_1 - p_2$	54.4	164.1	311	45	61	81.5	41.1	49.8	445	10,917	7	17,652	(76.4,83.3)	0.0	383.4
CMT9-25-$d_1 - \alpha_2 - p_1$	54.4	164.1	187	17	39	81.4	26.6	49.6	1246	28,703	12	33,724	(76.4,83.3)	0.0	1282.8
CMT9-25-$d_1 - \alpha_2 - p_2$	54.4	164.1	187	17	39	81.4	28.7	49.6	684	15,490	14	17,357	(70.6,83.3)	0.0	561.9
CMT9-25-$d_2 - \alpha_1 - p_1$	54.4	164.1	217	16	63	81.7	38.8	50.2	1181	32,238	12	48,954	(80.0,57.1)	0.0	1605.0
CMT9-25-$d_2 - \alpha_1 - p_2$	54.4	164.1	217	16	63	81.7	38.4	50.2	547	17,791	18	27,081	(86.7,57.1)	0.0	655.7
CMT9-25-$d_2 - \alpha_2 - p_1$	50.4	157.6	314	18	94	76.5	66.9	51.8	638	24,200	18	28,243	(86.7,85.7)	0.0	830.5
CMT9-25-$d_2 - \alpha_2 - p_2$	50.4	157.0	365	50	81	76.6	139.8	52.0	284	11,809	9	20,593	(86.7,85.7)	0.0	490.5
CMT9-25-$d_3 - \alpha_1 - p_1$	48.0	158.9	240	15	52	75.6	58.3	57.5	793	26,220	7	39,524	(83.3,75.0)	0.0	1115.6
CMT9-25-$d_3 - \alpha_1 - p_2$	48.0	158.9	240	15	52	75.6	58.6	57.5	893	43,918	7	65,663	(83.3,62.5)	0.0	2441.1
CMT9-25-$d_3 - \alpha_2 - p_1$	52.8	156.0	164	14	29	75.8	24.0	43.6	491	12,320	7	17,485	(83.3,62.5)	0.0	369.7
CMT9-25-$d_3 - \alpha_2 - p_2$	50.4	155.7	291	26	68	76.1	53.3	51.0	317	10,005	7	16,885	(83.3,87.5)	0.0	313.2
CMT9-30-$d_1 - \alpha_1 - p_1$	98.6	191.3	672	88	127	126.0	138.2	27.8	2156	53,287	23	56,097	(71.4,57.1)	7.8	TL
CMT9-30-$d_1 - \alpha_1 - p_2$	98.1	191.3	672	88	127	126.0	138.5	28.4	2180	53,116	22	52,877	(76.2,71.4)	22.7	TL
CMT9-30-$d_1 - \alpha_2 - p_1$	97.3	188.0	727	54	150	123.9	149.0	27.3	1656	53,944	19	52,571	(66.7,57.1)	48.2	TL
CMT9-30-$d_1 - \alpha_2 - p_2$	97.7	187.3	852	111	150	123.5	225.4	26.4	1450	56,617	12	56,823	(71.4,85.7)	18.2	TL
CMT9-30-$d_2 - \alpha_1 - p_1$	106.3	191.3	653	84	135	127.2	94.3	19.7	2528	48,843	11	47,137	(66.7,66.7)	37.3	TL
CMT9-30-$d_2 - \alpha_1 - p_2$	101.1	191.3	653	84	135	127.2	93.6	25.8	1732	51,083	16	51,762	(72.2,77.8)	18.2	TL
CMT9-30-$d_2 - \alpha_2 - p_1$	95.7	186.3	416	49	82	119.9	117.3	25.3	1404	50,481	13	49,272	(66.7,88.9)	42.2	TL
CMT9-30-$d_2 - \alpha_2 - p_2$	92.5	184.1	349	42	75	119.7	91.8	29.4	697	29,506	29	49,040	(66.7,77.8)	0.0	2882.4
CMT9-30-$d_3 - \alpha_1 - p_1$	92.5	186.1	339	30	60	117.1	69.2	26.6	802	27,702	23	29,709	(80.0,90.0)	0.0	1958.0
CMT9-30-$d_3 - \alpha_1 - p_2$	92.5	186.1	339	30	60	117.1	68.9	26.6	1284	39,195	27	39,362	(80.0,90.0)	0.0	3695.9
CMT9-30-$d_3 - \alpha_2 - p_1$	92.5	183.6	312	43	91	117.9	90.5	27.5	774	25,176	11	44,080	(80.0,90.0)	0.0	2375.2
CMT9-30-$d_3 - \alpha_2 - p_2$	89.8	183.0	780	73	150	118.0	204.6	31.4	1680	48,771	14	48,735	(86.7,80.0)	0.0	6628.3
Average								**44.1**						**9.2**	**2197.8**
#Optima														**29/36**	

Table A11
Computational results for the 36 instances produced by CMT10.

Name	Opt	pre-UB	#r-PI	#r-CI	#r-MTI	r-UB	r-time (s)	%-UB	#CI	#g-SEC	#g-LSEC	nodes	%-visited	%-gap	Time (s)
CMT10-20-$d_1-\alpha_1-p_1$	73.6	125.8	118	15	14	81.6	2.5	10.9	21	54	6	27	(57.1,60.0)	0.0	2.6
CMT10-20-$d_1-\alpha_1-p_2$	73.6	125.8	87	15	18	81.6	2.1	10.9	33	78	5	51	(57.1,60.0)	0.0	2.4
CMT10-20-$d_1-\alpha_2-p_1$	67.8	120.8	52	7	7	75.4	1.5	11.2	15	37	2	27	(57.1,80.0)	0.0	1.6
CMT10-20-$d_1-\alpha_2-p_2$	67.8	118.7	71	9	22	74.5	2.6	9.9	5	27	4	25	(57.1,80.0)	0.0	2.7
CMT10-20-$d_2-\alpha_1-p_1$	73.6	125.8	77	13	14	81.1	2.4	10.2	17	44	11	23	(50.0,60.0)	0.0	2.5
CMT10-20-$d_2-\alpha_1-p_2$	73.6	125.8	77	13	14	81.1	2.4	10.2	17	44	11	23	(50.0,50.0)	0.0	2.5
CMT10-20-$d_2-\alpha_2-p_1$	63.4	121.8	64	14	13	70.2	2.3	10.7	17	46	3	23	(50.0,66.7)	0.0	2.5
CMT10-20-$d_2-\alpha_2-p_2$	63.4	118.7	133	15	38	70.0	5.3	10.4	5	33	9	18	(58.3,83.3)	0.0	5.4
CMT10-20-$d_3-\alpha_1-p_1$	65.9	111.6	54	9	8	72.1	1.5	9.4	44	78	2	37	(40.0,71.4)	0.0	1.8
CMT10-20-$d_3-\alpha_1-p_2$	65.9	111.6	54	9	8	72.1	1.5	9.4	28	63	3	29	(40.0,71.4)	0.0	1.7
CMT10-20-$d_3-\alpha_2-p_1$	57.0	111.6	76	13	21	66.5	2.4	16.7	20	111	5	120	(40.0,85.7)	0.0	2.8
CMT10-20-$d_3-\alpha_2-p_2$	57.0	111.6	86	12	12	69.7	2.1	22.3	8	35	2	24	(50.0,85.7)	0.0	2.2
CMT10-25-$d_1-\alpha_1-p_1$	144.3	208.6	270	29	35	161.9	14.8	12.2	134	1095	8	1913	(64.7,66.7)	0.0	35.2
CMT10-25-$d_1-\alpha_1-p_2$	144.3	208.6	270	29	35	161.9	15.6	12.2	134	1095	8	1913	(64.7,66.7)	0.0	36.1
CMT10-25-$d_1-\alpha_2-p_1$	137.1	208.6	686	20	126	156.1	40.6	13.9	261	2032	14	2313	(70.6,83.3)	0.0	80.8
CMT10-25-$d_1-\alpha_2-p_2$	137.1	208.6	480	26	98	155.9	28.4	13.7	196	1196	18	1154	(64.7,66.7)	0.0	51.3
CMT10-25-$d_2-\alpha_1-p_1$	144.3	208.6	211	15	26	162.7	12.2	12.8	258	2109	21	2171	(53.3,71.4)	0.0	53.9
CMT10-25-$d_2-\alpha_1-p_2$	144.3	208.6	211	15	26	162.7	11.8	12.8	218	1439	14	1700	(60.0,71.4)	0.0	40.0
CMT10-25-$d_2-\alpha_2-p_1$	137.1	208.6	869	22	150	157.6	49.1	15.0	288	2099	9	2492	(66.7,85.7)	0.0	89.9
CMT10-25-$d_2-\alpha_2-p_2$	134.7	208.6	317	23	95	157.4	32.5	16.9	175	1394	26	1637	(66.7,85.7)	0.0	59.8
CMT10-25-$d_3-\alpha_1-p_1$	137.1	205.8	191	17	31	150.3	13.3	9.6	97	520	10	816	(75.0,62.5)	0.0	21.7
CMT10-25-$d_3-\alpha_1-p_2$	137.1	205.8	191	17	31	150.3	12.1	9.6	97	520	10	816	(75.0,62.5)	0.0	20.5
CMT10-25-$d_3-\alpha_2-p_1$	119.1	203.3	780	86	150	146.6	42.8	23.1	426	7040	1	9084	(50.0,87.5)	0.0	233.5
CMT10-25-$d_3-\alpha_2-p_2$	118.3	203.3	438	42	84	143.3	24.2	21.1	245	2844	9	3544	(50.0,75.0)	0.0	92.4
CMT10-30-$d_1-\alpha_1-p_1$	175.3	258.8	661	40	150	197.3	60.8	12.5	1041	11,593	15	13,826	(66.7,57.1)	0.0	830.1
CMT10-30-$d_1-\alpha_1-p_2$	175.3	258.8	661	40	150	197.3	63.1	12.5	1225	14,119	18	12,907	(61.9,42.9)	0.0	1008.9
CMT10-30-$d_1-\alpha_2-p_1$	164.2	258.8	908	81	150	193.7	59.3	18.0	2397	46,409	30	58,648	(66.7,85.7)	3.0	TL
CMT10-30-$d_1-\alpha_2-p_2$	167.2	258.8	915	46	150	194.4	65.7	16.3	1683	28,978	20	40,363	(66.7,85.7)	0.0	3363.6
CMT10-30-$d_2-\alpha_1-p_1$	169.2	258.8	297	23	85	198.7	34.0	17.4	1157	23,665	15	38,583	(55.6,77.8)	0.0	2875.0
CMT10-30-$d_2-\alpha_1-p_2$	169.2	258.8	297	23	85	198.7	35.1	17.4	1303	21,639	18	34,276	(55.6,77.8)	0.0	2474.3
CMT10-30-$d_2-\alpha_2-p_1$	168.5	258.8	1054	26	150	194.9	71.8	15.7	1300	17,055	24	20,310	(61.1,88.9)	0.0	1495.4
CMT10-30-$d_2-\alpha_2-p_2$	167.2	258.8	950	34	150	193.7	64.0	15.8	1481	20,577	13	27,684	(55.6,77.8)	0.0	2057.4
CMT10-30-$d_3-\alpha_1-p_1$	164.3	255.9	712	58	150	187.0	61.6	13.8	769	10,972	15	12,585	(66.7,70.0)	0.0	826.7
CMT10-30-$d_3-\alpha_1-p_2$	164.3	255.9	712	58	150	187.0	61.8	13.8	755	11,379	10	14,647	(66.7,70.0)	0.0	859.0
CMT10-30-$d_3-\alpha_2-p_1$	158.9	255.9	858	25	150	186.4	62.3	17.3	1575	32,249	11	42,890	(66.7,90.0)	0.0	4082.8
CMT10-30-$d_3-\alpha_2-p_2$	158.9	255.9	858	25	150	186.4	61.5	17.3	1149	18,204	17	26,724	(66.7,90.0)	0.0	1791.7
Average								**14.4**						**0.1**	**825.3**
#Optima														**35/36**	

the LSP in case of missing the SLR of the type *b* customer. The meaning of the remaining column headings is as follows:

References

[1] Aráoz J, Fernández E, Meza O. Solving the prize-collecting rural postman problem. Eur J Oper Res 2009;196(3):886–96.
[2] Archetti C, Bertazzi L, Laganà D, Vocaturo F. The undirected capacitated general routing problem with profits. Eur J Oper Res 2017;257(3):822–33.
[3] Archetti C, Bianchessi N, Speranza MG. Optimal solutions for routing problems with profits. Discrete Appl Math 2013;161(4):547–57.
[4] Archetti C, Bianchessi N, Speranza MG, Hertz A. The split delivery capacitated team orienteering problem. Networks 2014;63(1):16–33.
[5] Archetti C, Feillet D, Hertz A, Speranza MG. The capacitated team orienteering and profitable tour problems. J Oper Rese Soc 2009;60(6):831–42.
[6] Archetti C, Speranza MG. Arc routing problems with profits. In: Arc routing: problems, methods, and applications. In: MOS-SIAM series on optimization; 2013. p. 257–84.
[7] Archetti C, Speranza MG, Corberán Á, Sanchis JM, Plana I. The team orienteering arc routing problem. Transp Sci 2013;48(3):442–57.
[8] Archetti C, Speranza MG, Vigo D. Chapter 10: vehicle routing problems with profits. In: Vehicle routing: problems, methods, and applications. SIAM; 2014. p. 273–97.
[9] Augerat Ph, Belenguer JM, Benavent E, Corberán A, Naddef D, Rinaldi G. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report 949-M. Grenoble, France: Université Joseph Fourier; 1995.
[10] Balas E. The prize collecting traveling salesman problem. Networks 1989;19(6):621–36.
[11] Barahona F, Grötschel M. On the cycle polytope of a binary matroid. J Comb Theory Ser B 1986;40(1):40–62.
[12] Benavent E, Corberán A, Plana I, Sanchis JM. Min-max k-vehicles windy rural postman problem. Networks 2009;54(4):216–26.
[13] Benavent E, Corberán A, Plana I, Sanchis JM. New facets and an enhanced branch-and-cut for the min–max k-vehicles windy rural postman problem. Networks 2011;58(4):255–72.
[14] Bienstock D, Goemans MX, Simchi-Levi D, Williamson D. A note on the prize collecting traveling salesman problem. Math Program 1993;59(1):413–20.
[15] Bulhões T, Hà MH, Martinelli R, Vidal T. The vehicle routing problem with service level constraints. Eur J Oper Res 2018;265(2):544–58.
[16] Butt SE, Cavalier TM. A heuristic for the multiple tour maximum collection problem. Computers & Operations Research 1994;21(1):101–11.
[17] Chao I-M, Golden BL, Wasil EA. The team orienteering problem. Eur J Oper Res 1996;88(3):464–74.
[18] Christofides N, Mingozzi A, Toth P. The Vehicle Routing Problem. In: Christofides N, Mingozzi A, Toth P, Sandi C, editors. Combinatorial Optimisation. Chichester: Wiley; 1979. p. 315.
[19] Dell'Amico M, Maffioli F, Värbrand P. On prize-collecting tours and the asymmetric travelling salesman problem. Int Trans Oper Res 1995;2(3):297–308.
[20] Dikas G, Minis I. Scheduled paratransit transport systems. Transp Res Part B: Methodol 2014;67:18–34.
[21] Feillet D, Dejax P, Gendreau M. Traveling salesman problems with profits. Transp Sci 2005;39(2):188–205.
[22] Gendreau M, Laporte G, Semet F. A branch-and-cut algorithm for the undirected selective traveling salesman problem. Networks 1998;32(4):263–73.
[23] Goemans MX, Williamson DP. A general approximation technique for constrained forest problems. SIAM J Comput 1995;24(2):296–317.
[24] Golden BL, Levy L, Vohra R. The orienteering problem. Nav Res Logist 1987;34(3):307–18.
[25] Gunawan A, Lau HC, Vansteenwegen P. Orienteering problem: A survey of recent variants, solution approaches and applications. European Journal of Operational Research 2016;255(2):315–32.
[26] Hensher DA, Houghton E. Performance-based quality contracts for the bus sector: delivering social and commercial value for money. Transp Res Part B: Methodol 2004;38(2):123–46.
[27] Ibarra-Rojas OJ, Giesen R, Rios-Solis YA. An integrated approach for timetabling and vehicle scheduling problems to analyze the trade-off between level of service and operating costs of transit networks. Transp Res Part B: Methodol 2014;70:35–46.
[28] Jepsen MK, Petersen B, Spoorendonk S, Pisinger D. A branch-and-cut algorithm for the capacitated profitable tour problem. Discrete Optim 2014;14:78–96.
[29] Ke L, Zhai L, Li J, Chan FT. Pareto mimic algorithm: an approach to the team orienteering problem. Omega (Westport) 2016;61:155–66.
[30] Keshtkaran M, Ziarati K, Bettinelli A, Vigo D. Enhanced exact solution methods for the team orienteering problem. Int J Prod Res 2016;54(2):591–601.
[31] Krauth E, Moonen H, Popova V, Schut MC. Performance measurement and control in logistics service providing. In: ICEIS (2); 2005. p. 239–47.
[32] Li K, Tian H. A two-level self-adaptive variable neighborhood search algorithm for the prize-collecting vehicle routing problem. Appl Soft Comput 2016;43:469–79.
[33] Lysgaard J, Letchford AN, Eglese RW. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Math Program 2004;100(2):423–45.
[34] Moons K, Waeyenbergh G, Pintelon L. Measuring the logistics performance of internal hospital supply chains–a literature study. Omega **82** 2018:205–217.
[35] Padberg M, Rinaldi G. Facet identification for the symmetric traveling salesman polytope. Math Program 1990;47(1–3):219–57.
[36] Padberg MW, Rao MR. Odd minimum cut-sets and b-matchings. Math Oper Res 1982;7(1):67–80.
[37] Ralphs TK, Kopman L, Pulleyblank WR, Trotter LE. On the capacitated vehicle routing problem. Math Program 2003;94(2–3):343–59.
[38] Sawik T. On the fair optimization of cost and customer service level in a supply chain under disruption risks. Omega (Westport) 2015;53:58–66.
[39] Sherali HD, Smith JC. Improving discrete model representations via symmetry considerations. Manage Sci 2001;47(10):1396–407.
[40] Souffriau W, Vansteenwegen P, Berghe GV, Van Oudheusden D. The planning of cycle trips in the province of east flanders. Omega (Westport) 2011;39(2):209–13.
[41] Tang L, Wang X. Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem. Int J Adv Manuf Technol 2006;29(11):1246–58.
[42] Tarantilis CD, Stavropoulou F, Repoussis PP. The capacitated team orienteering problem: a bi-level filter-and-fan method. Eur J Oper Res 2013;224(1):65–78.
[43] Toth P, Vigo D. Vehicle routing: problems, methods, and applications. MOM-SIAM Series on Optimization, 18; 2014.
[44] Vansteenwegen P, Souffriau W, Van Oudheusden D. The orienteering problem: a survey. Eur J Oper Res 2011;209(1):1–10.
[45] Vidal T, Maculan N, Ochi LS, Vaz Penna PH. Large neighborhoods with implicit customer selection for vehicle routing problems with profits. Transp Sci 2015;50(2):720–34.