Computers & Operations Research 76 (2016) 226–237

Contents lists available at ScienceDirect

Computers & Operations Research

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

# Time constrained maximal covering salesman problem with weighted demands and partial coverage

Gizem Ozbaygin, Hande Yaman, Oya Ekin Karasan*

*Department of Industrial Engineering, Bilkent University, Ankara 06800, Turkey*

ARTICLE INFO

*Article history:*
Received 9 December 2015
Received in revised form
25 June 2016
Accepted 29 June 2016
Available online 30 June 2016

*Keywords:*
Covering salesman
Valid inequalities
Branch-and-cut

ABSTRACT

In a routing framework, it may not be viable to visit every single customer separately due to resource limitations or efficiency concerns. In such cases, utilizing the notion of coverage; i.e., satisfying the demand of multiple customers by visiting a single customer location, may be advantageous. With this motivation, we study the time constrained maximal covering salesman problem (TCMCSP) in which the aim is to find a tour visiting a subset of customers so that the amount of demand covered within a limited time is maximized. We provide flow and cut formulations and derive valid inequalities. Since the connectivity constraints and the proposed valid inequalities are exponential in the size of the problem, we devise different branch-and-cut schemes. Computational experiments performed on a set of problem instances demonstrate the effectiveness of the proposed valid inequalities in terms of strengthening the linear relaxation bounds as well as speeding up the solution procedure. Moreover, the results indicate the superiority of using a branch-and-cut methodology over a flow-based formulation. Finally, we discuss the relation between the problem parameters and the structure of optimal solutions based on the results of our experiments.

## 1. Introduction

The traveling salesman problem (TSP), one of the most intensively studied combinatorial optimization problems, aims to identify a least cost Hamiltonian tour on a given network. All nodes of the network must be visited exactly once in the TSP. But, this may not be viable in many real life applications due to resource limitations. Hence, identifying a tour over a subset of the nodes so that the others are within a reasonable distance of some tour stop can be more desirable. For instance, consider the postal delivery services. In a region with many demand points (customers), it may be very costly and inconvenient to visit every single demand point separately. Instead, positioning postboxes at a subset of customer locations and collecting mails through these boxes is more efficient. In such a system, each postbox is used for covering the demand of multiple customers; i.e., the customers without a postbox at their location can drop off mails to their closest box. A relevant objective in this context would be the minimization of tour length (cost). However, there may also be cases where cost is not the primary concern. Consider the routing of mobile health facilities as an example. The main issue here is to ensure, with the available resources, that the number of patients who receive health care is maximized.

In this study, we consider the time constrained maximal covering salesman problem (TCMCSP), in which the aim is to find a tour visiting a subset of the demand points, so as to maximize the demand covered subject to a time constraint. We assume that the demands of the vertices that are on the tour are fully covered while only a certain percentage $\alpha$ of the demand of a vertex is covered if it is not visited but is within a specified distance $r$ of some tour stop. This is a realistic assumption because not every demand point may be willing to travel a distance of $r$ units to reach a tour stop. As an example, a passenger may not want to take the bus to his destination if he does not want to walk to the closest bus stop because he is closer to a metro stop. Thus, we can presume that $\alpha$% of the demands is covered regarding the points not on the tour. Due to the upper bound on the tour duration, some vertices may be left isolated (uncovered); i.e., they are not served at all.

Potential real-life applications of TCMCSP include the routing of mobile health facilities, collection of blood, distribution of food, drinking water and medical supplies in the aftermath of a disaster, routing of security patrol cars in rural regions for crime prevention, routing of unmanned aerial vehicles (UAVs) for information gathering against intruders, deciding on the order and the location of the meetings for a political campaign and so on. In all of these

* Corresponding author.
*E-mail address:* karasan@bilkent.edu.tr (O.E. Karasan).

http://dx.doi.org/10.1016/j.cor.2016.06.019
0305-0548/© 2016 Elsevier Ltd. All rights reserved.

applications, there is a restriction on the tour length, and the primary objective is to maximize coverage.

To the best of our knowledge, only a single study exists on the TCMCSP (due to [1]). The problem structure in this study is different from ours since the authors in [1] distinguish the set of customers and the set of facilities, and they identify a tour over a subset of the facilities. In addition, they consider the allocation of customers to facilities. The problem we study is a special case of the one in [1] where the set of facilities and the set of customers are the same and the allocation is disregarded. For this special case, we are able to propose stronger and smaller formulations and efficient solution methods.

Our contributions can be summarized as follows. We study the TCMCSP with weighted demands and partial coverage. We propose two mathematical formulations and valid inequalities for the problem, and develop branch-and-cut solution methodologies. We are able to solve instances of realistic sizes to optimality within a time limit of one hour.

The rest of this paper is organized as follows. The literature on several problems that are related to TCMCSP are reviewed in the next section. Mathematical formulations and valid inequalities for TCMCSP are given in Section 3. Section 4 presents four branch-and-cut schemes to solve the problem. The results of our computational study along with a discussion of the sensitivity of optimal solutions to changes in problem parameters are reported in Section 5. Finally, Section 6 concludes the paper with a summary of our findings.

## 2. Related literature

The first problem incorporating the coverage concept into a routing scheme is the covering salesman problem (CSP). CSP is the problem of identifying a minimum length Hamiltonian tour over a subset of vertices in a way that every vertex not on the tour lies within a certain distance of some visited vertex. The CSP is formally introduced in [2] where a heuristic algorithm is proposed to solve the problem. Later, the geometric version of the CSP is studied in [3] and polynomial time approximation algorithms are presented with a bounded error ratio regarding the optimal tour length.

Two multi-objective variants of the CSP are considered in [4]. These are the median tour problem (MTP) and the maximal covering tour problem (MCTP) where the tour should visit a predetermined number of vertices and the objectives are: (1) minimization of the tour length and (2) maximization of the accessibility to the tour for the vertices that are not visited. A heuristic approach is suggested to approximate the frontier of the efficient solutions.

A generalization of the CSP is studied in [5] and [6] where an additional cost is incurred for every node visited by the tour and each node is associated with a weighted demand representing the minimum number of times it has to be covered. Another generalization of the CSP, called the generalized covering traveling salesman problem (GCTSP), is presented in [7]. In the GCTSP, one aims to find a minimum length tour passing through a subset of facilities while covering at least a predetermined number of customers. Node-based and flow-based formulations are presented and two metaheuristic approaches are developed for the problem. The TCMCSP is introduced in [1], where the goal is to maximize the number of covered customers with an upper bound on the total traveling time. In a sense, it is complementary to the GCTSP. As pointed out earlier, the problem we study in this paper is a special case of that considered in [1].

A very popular generalization of the CSP is the covering tour problem (CTP) introduced in [8]. Given an undirected graph $G = (V \cup W, E)$, the CTP is the problem of identifying a minimum length Hamiltonian tour in which the vertices in $T \subset V$ must be on the tour while the remaining vertices in $V$ may or may not be visited, and the vertices in $W$ should be covered without being visited.

The problem of planning mobile healthcare facilities in Suhum District of Ghana is modeled as the CTP in [9] and solved with the algorithm developed in [8]. In [10], a GRASP is devised for solving a generalization of the CTP in which the vertices in $W$ can also be visited. A two-commodity flow formulation and three scatter search methods for the CTP are presented in [11]. Several other heuristics are proposed in [12].

The multi-vehicle variant of the CTP ($m$-CTP) is introduced in [13]. For each tour, there is an upper bound on its length and an upper bound on the number of vertices visited. The $m$-CTP is formulated as an integer linear program using vehicle flow variables and heuristic algorithms are developed. A covering tour perspective is adopted in [14] to tackle the problem of locating satellite distribution centers to supply humanitarian aid over a disaster area. The problem of planning routes for routine patrol cars is also modeled as the $m$-CTP in [15].

The CTP is investigated in a multi-objective setting as well. In [16], the bi-objective CTP (BOCTP) is introduced, and a multi-objective evolutionary algorithm is proposed. A post-natural-disaster related problem is addressed in [17] in which part of the infrastructure in the region affected by the disaster is destroyed. The problem of supplying food, medicine and shelter over the affected region is considered as a multi-objective CTP and heuristics are presented to solve the problem. A variant of the BOCTP with stochastic demands is introduced in [18] and modeled as a two-stage stochastic program with recourse, which is solved using an epsilon-constraint approach involving branch-and-cut.

The TCMCSP is also related to traveling salesman problems with profits, which are classified into three categories in [19] based on their objectives. They are (1) maximizing profit under a distance constraint, (2) minimizing distance under a profit constraint and (3) a combination of distance minimization and profit maximization. The TCMCSP is closest to the problems in class (1), which also contains the orienteering problem (OP) introduced in [20]. In the OP, also known as the selective traveling salesman problem [21] or the maximum collection problem [22], every vertex is associated with a profit and the objective is to find a tour with maximum profit subject to a time restriction. The OP is a special case of the TCMCSP with $r=0$; that is, the demand of a vertex is covered only if it is visited. We refer the interested reader to [23] for a recent survey regarding OP.

## 3. Formulation and valid inequalities

In this section, we formally define TCMCSP and propose two mathematical models. Afterwards, we present several classes of valid inequalities for the problem.

Let $G = (V, E)$ be an undirected complete graph with the set of vertices $V = \{0, 1, \ldots, n\}$ and the set of edges $E = \{\{i, j\}: i, j \in V, i < j\}$ . Suppose that vertex 0 represents a central depot and the remaining vertices correspond to demand points. We assume that the tour contains at least three vertices and that $|V| \geq 4$. Every edge $e \in E$ is associated with a nonnegative length $c_e$ and every vertex $i \in V \setminus \{0\}$ has a positive demand denoted by $d_i$. We assume that the edge lengths satisfy the triangle inequality. Define $N_i$ to be the set of vertices within the coverage distance $r_i$ of vertex $i$ other than the vertex $i$ itself, i.e., $N_i = \{j \in V: c_{\{i,j\}} \leq r_i, j \neq i\}$. The time constrained maximal covering salesman problem is the problem of determining a route over a subset of vertices in $V$ that maximizes the total demand covered. The route must start and end at the depot and its length should not exceed a prespecified threshold value $L$. A vertex $i$ can be either visited or covered by some vertex $j \in N_i$, or left isolated. We consider the demand of the vertex $i$ as fully covered if it is on the tour while only a percentage $\alpha > 0$ of its demand can be covered if $i$ is not visited and the tour contains at least one vertex from the set $N_i$.

### 3.1. Mathematical formulations

Our first model is a cut based model. We use the following variables in this model:

$$x_e = \begin{cases} 1 \text{ if edge } e \in E \text{ is on the tour,} \\ 0 \text{ otherwise,} \end{cases}$$

$$y_i = \begin{cases} 1 \text{ if vertex } i \in V \text{ is on the tour,} \\ 0 \text{ otherwise,} \end{cases}$$

$$z_i = \begin{cases} 1 \text{ if vertex } i \in V \text{ is not visited but covered by some vertex in } N_i, \\ 0 \ \text{ otherwise.} \end{cases}$$

For $S \subset V$, we let $\delta(S) = \{e \in E\colon |e \cap S| = 1\}$ and $E(S) = \{e \in E\colon |e \cap S| = 2\}$ . If $S = \{i\}$ we simply use $\delta(i)$ instead of $\delta(\{i\})$. Finally, we write $x(E') = \sum_{e \in E'} x_e$ for $E' \subseteq E$ and $y(S) = \sum_{i \in S} y_i$ for $S \subseteq V$. Our first mathematical model for the TCMCSP is given below:

$$\max \sum_{i \in V \setminus \{0\}} d_i(y_i + \alpha z_i), \tag{1}$$

$$\text{s. t.} \quad \sum_{e \in E} c_e x_e \le L, \tag{2}$$

$$y_i + z_i \le 1 \quad i \in V \setminus \{0\}, \tag{3}$$

$$x(\delta(i)) = 2y_i \quad i \in V, \tag{4}$$

$$z_i \le y(N_i) \quad i \in V \setminus \{0\}, \tag{5}$$

$$x(\delta(S)) \ge 2y_i \quad S \subset V \setminus \{0\}, 3 \le |S| \le n-2, i \in S, \tag{6}$$

$$y_0 = 1, \tag{7}$$

$$x_e \in \{0, 1\} \quad e \in E, \tag{8}$$

$$y_i \in \{0, 1\} \quad i \in V \setminus \{0\}, \tag{9}$$

$$z_i \in \{0, 1\} \quad i \in V \setminus \{0\}. \tag{10}$$

The objective (1) is to maximize the total demand covered. Constraint (2) ensures that the total tour length does not exceed $L$. Due to (3), a vertex can be isolated, on the tour or covered by the tour stops within its coverage distance. Degree requirement for each vertex is imposed by (4), i.e., a visited vertex has degree two whereas a vertex that is not visited cannot have any edge adjacent to it. Constraints (5) guarantee that a vertex $i$ cannot be covered if none of the vertices in $N_i$ is visited. Connectivity cuts in (6) prevent subtours and are exponential in the size of the problem. Constraint (7) enforces the depot to be on the tour. Finally, domain restrictions on the variables are given in (8)–(10).

Next, we modify the formulation presented in [1] to model our problem. This model is a flow based directed model. For this reason, we define the set of arcs $A = \{(i, j), (j, i)\colon \{i, j\} \in E\}$ and we let the length of arcs $\hat{c}_{ij} = \hat{c}_{ji} = c_{\{i,j\}}$ for each $\{i, j\} \in E$. In addition to the $y$ and $z$ variables defined above, we use the following decision variables: $\hat{x}_a$ is 1 if arc $a \in A$ is part of the tour and it is 0 otherwise. Also, $f_{ij}$ is the total traveled time (distance) from the depot to vertex $j$, when traversing arc $(i, j) \in A$. For $A' \subseteq A$, we let $\hat{x}(A') = \sum_{a \in A'} \hat{x}_a$ and $f(A') = \sum_{a \in A'} f_a$. Then, the TCMCSP can be modeled as:

$$\max \sum_{i \in V \setminus \{0\}} d_i(y_i + \alpha z_i), \tag{11}$$

$$\text{s. t.} \quad (3), (5), (7), (9), (10)$$

$$\sum_{a \in A} \hat{c}_a \hat{x}_a \le L, \tag{12}$$

$$\hat{x}(\delta^+(i)) = \hat{x}(\delta^-(i)) = y_i \qquad i \in V, \tag{13}$$

$$f(\delta^+(i)) - f(\delta^-(i)) = \sum_{a \in \delta^+(i)} \hat{c}_a \hat{x}_a \quad i \in V \setminus \{0\}, \tag{14}$$

$$f_{0i} = \hat{c}_{0i} \hat{x}_{0i} \quad i \in V \setminus \{0\}, \tag{15}$$

$$f_{ij} \le (L - \hat{c}_{j0}) \hat{x}_{ij} \quad (i, j) \in A, j \ne 0, \tag{16}$$

$$f_{i0} \le L \hat{x}_{i0} \quad i \in V \setminus \{0\}, \tag{17}$$

$$f_{ij} \ge (\hat{c}_{0i} + \hat{c}_{ij}) \hat{x}_{ij} \quad (i, j) \in A, i \ne 0, \tag{18}$$

$$\hat{x}_a \in \{0, 1\}, f_a \ge 0 \quad a \in A. \tag{19}$$

Here constraints (13) are degree constraints. Constraints (14)–(18) relate the variables $f$'s and $\hat{x}$'s and eliminate subtours. Note that this formulation is valid only for instances with positive arc lengths. Otherwise, it is possible to obtain solutions that contain zero length subtours and the nodes on these subtours are counted as visited.

### 3.2. Lifting connectivity constraints

Our initial computational experiments (presented in Section 4) show that we are able to solve larger instances using the first model based on connectivity cuts. In the sequel, we present valid inequalities for the feasible set of this model that we denote by $Y$.

For $\hat{V} \subseteq V$, we define $Y^0(\hat{V}) = \{(x, y, z) \in Y\colon z_j = 0, j \in \hat{V}\}$. $Y^0(V)$ is the feasible set of an orienteering problem. In other words, the polytope of the orienteering problem is a face of the polytope of TCMCSP where all $z_j$ variables are fixed to zero. The family of valid inequalities that we present is obtained by lifting the connectivity cuts with variables $z_i$'s. These inequalities are strong when the connectivity cut that is lifted is strong for the polytope associated with the orienteering problem and some mild conditions are satisfied.

**Theorem 3.1.** *Let $S \subset V\setminus\{0\}$ with $3 \le |S| \le n-2$ and $i \in S$ such that $\emptyset \ne N_i \subset S$. The lifted connectivity inequality (LCI)*

$$x(\delta(S)) \ge 2y_i + 2z_i \tag{20}$$

*is valid for Y.*

**Proof.** If $z_i=0$, then (20) reduces to constraint (6). If $z_i=1$, then $y_i=0$ and $y(N_i) \ge 1$. Since $N_i \subset S$ and $y(N_i) \ge 1$, there exists $j \in S$ with $y_j=1$. Constraint (6) for set $S$ and node $j$ implies $x(\delta(S)) \ge 2$. Hence, inequality (20) is satisfied in both cases. □

Let $V' = \{j \in V\setminus\{0\}: N_j \ne \emptyset\}$. If $j \notin V'$, then we know that $z_j=0$ in all feasible solutions. Hence $Y^0(V) = Y^0(V')$.

**Theorem 3.2.** *Suppose that the connectivity constraint (6) for set $S \subset V\setminus\{0\}$ with $3 \le |S| \le n-2$ and node $i \in S \cap V'$ is facet defining for $\mathrm{conv}(Y^0(V))$, $N_i \subset S$ and the cycle on nodes $\{0, k, l\}$ satisfies constraint (2) for any two distinct nodes k and l in $V\setminus\{0\}$. Then inequality (20) is facet defining for $\mathrm{conv}(Y)$.*

**Proof.** We lift the connectivity constraint (6) first with $z_i$ and then with $z_j$ for $j \in V'\setminus\{i\}$. We first would like to find $\sigma_i$ such that the inequality $x(\delta(S)) \ge 2y_i + \sigma_i z_i$ is satisfied by all solutions in $Y^0(V'\setminus\{i\})$. The inequality is clearly satisfied for all $\sigma_i$ when $z_i=0$. If $z_i=1$, then $y_i=0$ and $y(N_i) \ge 1$. In this case, we need $\sigma_i \le x(\delta(S))$ for all $(x, y, z)$ in $Y^0(V'\setminus\{i\})$ with $z_i=1$. Equivalently, we want $\sigma_i \le \min_{(x,y,z)\in Y^0(V'\setminus\{i\}):z_i=1} x(\delta(S))$. We know that $\min_{(x,y,z)\in Y^0(V'\setminus\{i\}):z_i=1} x(\delta(S)) \ge 2$ since $N_i \subset S$ and $y(N_i) \ge 1$. Consider the solution where $z_i=1$, $y_i=0$, $y_k = y_l = 1$, $z_k = z_l = 0$ for some $k \in N_i$ and some $l \in V\setminus\{0, i, k\}$ and for all other nodes $j$, we have $y_j = z_j = 0$. We let $x_{\{0,k\}} = x_{\{k,l\}} = x_{\{l,0\}} = 1$ and other edge variables be zero. This solution is in $Y^0(V'\setminus\{i\})$ with $z_i=1$ and $x(\delta(S)) = 2$. Hence $\min_{(x,y,z)\in Y^0(V'\setminus\{i\}):z_i=1} x(\delta(S)) = 2$. Consequently, the inequality $x(\delta(S)) \ge 2y_i + \sigma_i z_i$ is valid for all $\sigma_i \le 2$ and the inequality $x(\delta(S)) \ge 2y_i + 2z_i$ is facet defining for $\mathrm{conv}(Y^0(V'\setminus\{i\}))$.

Now let $\pi$ be a permutation on $V'\setminus\{i\}$. Next we lift inequality $x(\delta(S)) \ge 2y_i + 2z_i$ with variables $z_l$ for $l \in V'\setminus\{i\}$ in the order $\pi$. To lift inequality $x(\delta(S)) \ge 2y_i + 2z_i$ with $z_{\pi(1)}$, we would like to compute $\min_{(x,y,z)\in Y^0(V'\setminus\{i,\pi(1)\}):z_{\pi(1)}=1}(x(\delta(S)) - 2y_i - 2z_i)$. We know that $x(\delta(S)) - 2y_i - 2z_i \ge 0$ for all $(x, y, z) \in Y^0(V'\setminus\{i, \pi(1)\})$. If $i \in N_{\pi(1)}$, then let $k$ be any node in $V\setminus\{0, i, \pi(1)\}$. If $i \notin N_{\pi(1)}$, then let $k$ be a node in $N_{\pi(1)}$. We set $z_{\pi(1)} = 1$, $y_{\pi(1)} = 0$, $y_k = y_i = 1$, $z_k = z_i = 0$ and for other nodes $j$, we have $y_j = z_j = 0$. We use edges $\{0, k\}$, $\{k, i\}$, and $\{i, 0\}$. This solution is in $Y^0(V'\setminus\{i, \pi(1)\})$ with $z_{\pi(1)} = 1$ and $x(\delta(S)) - 2y_i - 2z_i = 0$. Hence the optimal lifting coefficient for $z_{\pi(1)}$ is zero and the inequality $x(\delta(S)) \ge 2y_i + 2z_i$ is facet defining for $\mathrm{conv}(Y^0(V'\setminus\{i, \pi(1)\}))$. Now suppose that the inequality is facet defining for $\mathrm{conv}(Y^0(V'\setminus\{i, \pi(1), \ldots \pi(u-1)\}))$ for $2 \le u \le |V'|$ and that we are lifting it with $z_{\pi(u)}$. We can show using the same arguments that the optimal coefficient for $z_{\pi(u)}$ is also zero. Hence inequality (20) is facet defining for $\mathrm{conv}(Y)$. □

If $S = N_i \cup \{i\}$, then we call the resulting inequality (20) a simple lifted connectivity inequality (SLCI). Note that these inequalities are polynomial in number.

### 3.3. Optimality cuts and simple cover inequalities

We also derive some optimality cuts and simple valid inequalities for TCMCSP based on the idea of knapsack covers.

**Theorem 3.3.** *For $i \in V'$ such that $0 \in N_i$, in an optimal solution the following equality holds*

$$y_i + z_i = 1. \tag{21}$$

**Proof.** Since both $d_i$ and $\alpha$ are positive, the result follows. □

Hence, we use (21) instead of constraint (3) for such vertices.

**Theorem 3.4.** *Let i and j be distinct vertices in $V\setminus\{0\}$. If $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$, then every feasible solution satisfies $x_{ij} = 0$. In addition, the inequality $y_i + y_j \le 1$ holds.*

**Proof.** For two distinct vertices $i$ and $j$ satisfying the above condition, we have $x_{ij} = 0$ because we assume that $c$ satisfies the triangle inequality. Moreover, at most one of the vertices $i$ and $j$ can be visited in any feasible solution since even the length of a shortest cycle on {0, i, j} exceeds the bound $L$. Hence, the inequality $y_i + y_j \le 1$ is valid. □

**Theorem 3.5.** *Let $S \subset V\setminus\{0\}$ and $i, j \in S$ be two distinct vertices such that $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$. Then, the following cover inequality*

$$x(\delta(S)) \ge 2y_i + 2y_j \tag{22}$$

*is valid and dominates constraint (6) for both i and j.*

**Proof.** We know that at most one of $i$ and $j$ can be included in a feasible solution by the previous theorem. If $y_i=1$, then $y_j=0$ and (22) reduces to the connectivity constraint (6) for $S$ and $i$. The case for $y_j=1$ is similar. Hence, the result follows. □

## 4. Branch-and-cut algorithms

We devise branch-and-cut algorithms to solve the TCMCSP since our cut formulation involves exponentially many constraints. We propose four branch-and-cut schemes. The most basic version starts by solving the relaxation (1)–(5), (7)–(10) and the violated connectivity constraints (6) are introduced only for integer solutions of the branch-and-cut tree. In the second scheme, we separate the connectivity constraints also for fractional solutions at the root node of the tree. The last two schemes are similar to the second one in terms of where in the solution tree the separation procedures are executed. However, in the third scheme, we add the SLCIs for $i \in V\setminus\{0\}$ with $0 \notin N_i$ to the initial relaxation. Moreover, before checking whether a connectivity constraint $x(\delta(S)) \ge 2y_i$ is violated, we investigate the corresponding LCI given by $x(\delta(S)) \ge 2(y_i + z_i)$ if $0 \notin N_i$ and $N_i \subset S$. It may be the case that the condition $0 \notin N_i$ holds, yet there exists $j \in N_i\setminus S$. In that case, we extend the set $S$ by adding the vertices in $N_i\setminus S$ to it and explore whether the inequality $x(\delta(S \cup N_i)) \ge 2(y_i + z_i)$ is satisfied. If either of these LCIs is violated, we introduce it instead of the connectivity constraint. The last branch-and-cut scheme is similar to the third one. Besides adding SLCIs to the initial relaxation (1)–(5), (7)–(10), we add the inequalities $y_i + y_j \le 1$ and set $x_{ij} = 0$ for every vertex pair $i, j \in V\setminus\{0\}$ such that $i < j$ and $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$ . During the separation of connectivity constraints, we search for violated cover inequalities in addition to LCIs. For a set $S \subset V\setminus\{0\}$ and a vertex $i \in S$, if we detect $j \in S$ with $j \ne i$ for which the corresponding cover inequality is not satisfied or if we identify a violated LCI, we do not check the violation of the connectivity constraint induced by $i$ and $S$. Our separation procedures are described in detail in the next subsections.

### 4.1. Separation of connectivity constraints

In separating the connectivity constraints, we use the ideas proposed in [24]. Suppose that $\bar{G} = (\bar{V}, \bar{E})$ is the support graph induced by a given solution vector $(\bar{x}, \bar{y}, \bar{z})$; i.e., $\bar{V} = \{i \in V: \bar{y}_i > 0\}$ and $\bar{E} = \{e \in E: \bar{x}_e > 0\}$. Let $S_j$, $j = 0, 1, \ldots, t$ be the connected components of $\bar{G}$ where $0 \in S_0$. There are two possibilities regarding the solution vector $(\bar{x}, \bar{y}, \bar{z})$: either it is integral or it has at least one fractional component. In the former case, the solution is feasible for the TCMCSP if and only if $t=0$; that is, the corresponding support graph $\bar{G}$ is connected. When $t \geq 1$, there is a connectivity cut violated by $S_j$ and each $i \in S_j$ for every $j=1,\ldots,t$. Hence, the solution vector induces $\sum_{j=1}^{t} |S_j|$ violated constraints and introducing any one of them to the model cuts off the current solution. Nevertheless, instead of adding a single cut at a time, we add the cut (6) for every $S_j$, $j=1,\ldots,t$ and for every $i \in S_j$ in order to speed up the solution procedure.

Now consider the case where the solution is fractional. If $t \geq 1$ for the corresponding support graph $\bar{G}$, a violated connectivity constraint is induced by every $S_j$ and $i \in S_j$ for $j=1,\ldots,t$ as in the previous case. However, if $\bar{G}$ is connected, exact separation of violated connectivity cuts can be performed by solving a series of minimum cut problems on the graph $\bar{G}$. Checking violation of the inequality $x(\delta(S)) \geq 2\bar{y}_k$ for a vertex $k \in \bar{V} \setminus \{0\}$ and for every $S \subset V \setminus \{0\}$ such that $k \in S$ is equivalent to checking whether the capacity of a minimum cut separating the vertices $k$ and 0 is greater than or equal to $2\bar{y}_k$ when the capacity of each edge $e \in \bar{E}$ is set to $\bar{x}_e$. If the capacity of a minimum cut separating vertex $k$ and vertex 0 is at least $2\bar{y}_k$, then every $S \subset V \setminus \{0\}$ containing $k$ satisfies (6). Otherwise, we obtain a violated connectivity cut corresponding to $k$ and the vertex partition $S^*(k)$, where $[S^*(k), V \setminus S^*(k)]$ defines a minimum cut with respect to source $k$ and sink 0. In particular, $S^*(k)$ is the vertex partition containing $k$.

In some cases, there may be a more efficient way to identify violated connectivity cuts than solving $|\bar{V}| - 1$ minimum cut problems. Let $cap(S)$ denote the capacity of the cut defined by the set $S$ of vertices. Suppose that $S^* \subset \bar{V} \setminus \{0\}$ is the set of vertices inducing a global minimum cut of the graph $\bar{G}$. Then, we have $cap(S^*(k)) = cap(S^*)$ and thus, $S^*(k) = S^*$ for every $k \in S^*$. This means that $S^*$ is a minimum cut separating any $k \in S^*$ from the vertex 0, and we do not have to solve a separate minimum cut problem for each vertex in $S^*$. Besides, we know that $cap(S) \geq cap(S^*)$ for any cut $S$ of the graph $\bar{G}$. This implies that the corresponding connectivity constraints are satisfied for the vertices $i \in (\bar{V} \setminus \{0\}) \setminus S^*$ such that $cap(S^*) \geq 2\bar{y}_i$. Therefore, we can eliminate these vertices from consideration as well. For each of the remaining vertices, we solve a minimum cut problem to determine if there are any violated connectivity constraints.

### 4.2. Separation of lifted connectivity and cover inequalities

We investigate violated lifted connectivity and cover inequalities during the execution of connectivity constraint separation procedure. More specifically, for a particular vertex $i$ and a set $S \subset V \setminus \{0\}$ with $i \in S$, we separate LCIs and cover inequalities prior to the corresponding connectivity constraint. In the following, we describe our separation subroutines for LCIs and cover inequalities.

First, consider the case with an integer solution containing subtours denoted by $S_0, S_1, \ldots, S_t$ where $0 \in S_0$. Take any $S_k$ for $k=1, \ldots, t$. Let $i \in S_k$ and $j$ be a vertex such that $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$. Then $y_j=0$ must hold. Clearly, we have $x(\delta(S_k \cup \{j\})) = 0$ because both $x(\delta(S_k))$ and $x(\delta(j))$ are equal to zero. Now observe that the set $S_k \cup \{j\}$ and the vertices $i, j$ induce a violated cover inequality. Hence, while examining a certain vertex $i \in S_k$ for a particular $k = 1, \ldots, t$, we introduce a cover inequality for $i$ and each $j$ such that $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$. After exploring these inequalities, we evaluate whether the LCI induced by the vertex $i$ and the set $N_i \cup S_k$ is satisfied, provided that $0 \notin N_i$. If $N_i \subset S_k$, then we have a violated LCI since $x(\delta(S_k)) = 0$. Else, we should verify that $x(\delta(N_i \setminus S_k)) = 0$ to be able to add the corresponding LCI, as otherwise $x(\delta(N_i \cup S_k))$ is at least two and the inequality is satisfied. We add the connectivity constraint induced by $i$ and $S_k$ if no cover inequality involving the vertex $i$ is identified so far and either one of the following conditions holds:

- $0 \notin N_i$ but no LCI violation is detected,
- $0 \in N_i$.

Suppose now that we have a fractional solution. If the solution contains subtours $S_0, S_1, \ldots, S_t$ with $0 \in S_0$, we separate LCIs and connectivity constraints in the same manner as we do in the integer solution case. However, our cover inequality separation procedure is slightly different. For a subtour $S_k$ for $k=1,\ldots,t$ and a vertex $i \in S_k$, we add the cover inequalities for all $j \in S_k$ such that $i < j$ and $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$.

If the support graph associated with a given fractional solution is connected, we embed our search for LCI and cover inequalities into our connectivity constraint separation algorithm as well. Once a global minimum cut $S^*$ of the support graph $\bar{G}$ is determined, we check violation of the LCIs and cover inequalities for each $i \in S^*$ before exploring the corresponding connectivity constraint. The capacity of the cut $S^* \cup N_i$ is compared with the value $2(y_i + z_i)$ to investigate whether the LCI associated with $i$ and $S^* \cup N_i$ is satisfied (if $0 \notin N_i$). If it is violated, then we add it to the model. Afterwards, for every $j \in S^*$ such that $i < j$ and $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$, we introduce the cover inequality induced by $i$, $j$ and $S^*$ if the capacity of $S^*$ is less than $2(y_i + y_j)$. Upon completing the search for LCIs and cover inequalities for all $i \in S^*$, we evaluate violation of the connectivity constraint regarding each vertex in $S^*$ for which no violated LCI or cover inequality is identified.

Among the vertices in $\bar{V} \setminus (S^* \cup \{0\})$, we eliminate those with $2(y_k + z_k) \leq \bar{x}(\delta(S^*))$ because the capacity of any cut of $\bar{G}$ is at least as large as that of $S^*$. Hence, our algorithm will not detect any violated LCI or connectivity constraint for such vertices. For each remaining vertex $i$, we find the minimum cut $S^*(i)$ as in our connectivity constraint separation procedure, and check whether the LCI induced by $i$ and $S^*(i) \cup N_i$ is satisfied (again if $0 \notin N_i$). Then, we inspect every $j \in S^*(i)$ such that $j \neq i$ and $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$ in case a violated cover inequality exists. If the LCI and cover inequalities associated with the vertex $i$ are all found to be satisfied, we investigate the corresponding connectivity constraint.

### 4.3. Other implementation details

We invoke *StoerWagnerMinimumCut* and *MinSourceSinkCut* procedures from the Java graph theory (*jgrapht*) library to find a global minimum cut of a given undirected graph (it implements the algorithm in [25]) and a minimum cut between a specified pair of source and sink nodes in a directed graph (it implements the algorithm in [26]), respectively. Note that since *MinSourceSinkCut* works on a directed graph, we transform $\bar{G}$ into a directed graph by replacing each edge $\{i, j\} \in \bar{E}$ with the arcs $(i,j)$ and $(j,i)$, and by assigning a capacity of $\bar{x}_{ij}$ units to both arcs after solving the global minimum cut problem.

In our branch-and-cut schemes that involve separation for fractional solutions, all cuts (LCIs, cover inequalities, connectivity constraints) are separated only at the root node of the branch-and-cut tree. Obviously, the separation for integer solutions is conducted everywhere. For a fractional solution, we consider an inequality to be violated if its violation exceeds 5% based on the results of our preliminary experiments. The violation of an

**Table 1**
Results with the flow formulation.

| Solution times (s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | | | $r=10$ | | | $r=20$ | | |
| Name | $n$ | $\alpha$ | $L_1$ | $L_2$ | $L_3$ | $L_1$ | $L_2$ | $L_3$ |
| *p*01 | 51 | 0.50 | 99.57 | 23.83 | 21.28 | 39.65 | 7.57 | 3.23 |
| | | 0.75 | 266.89 | 47.21 | 30.07 | 102.28 | 19.63 | 3.69 |
| *p*02 | 76 | 0.50 | (5.72) | 91.20 | 92.06 | 1229.98 | 13.86 | 57.81 |
| | | 0.75 | (12.79) | 175.50 | 123.81 | (4.49) | 66.35 | 70.93 |
| *p*03 | 101 | 0.50 | (13.76) | 1842.29 | 291.56 | (5.56) | 267.11 | 213.54 |
| | | 0.75 | (24.64) | 2328.21 | 750.29 | (6.65) | 416.44 | 248.80 |
| *p*12 | 101 | 0.50 | (70.54) | (29.76) | (8.97) | (36.84) | (22.17) | (8.85) |
| | | 0.75 | (70.89) | (14.51) | (4.06) | (26.91) | (12.95) | (5.02) |
| Root gaps (%) | | | | | | | | |
| *p*01 | 51 | 0.50 | 21.57 | 3.65 | 1.23 | 8.02 | 1.89 | 0.26 |
| | | 0.75 | 25.65 | 4.02 | 0.65 | 5.42 | 1.07 | 0.21 |
| *p*02 | 76 | 0.50 | 21.99 | 3.34 | 1.77 | 8.39 | 0.47 | 0.82 |
| | | 0.75 | 26.33 | 2.97 | 0.71 | 6.56 | 0.47 | 0.34 |
| *p*03 | 101 | 0.50 | 21.73 | 3.20 | 0.84 | 10.84 | 1.68 | 0.49 |
| | | 0.75 | 21.66 | 1.58 | 0.49 | 6.58 | 0.88 | 0.32 |
| *p*12 | 101 | 0.50 | 87.56 | 33.06 | 14.82 | 48.58 | 30.45 | 14.92 |
| | | 0.75 | 76.50 | 19.03 | 6.94 | 42.09 | 15.17 | 6.94 |

inequality $LHS \geq RHS$, where LHS and RHS represent the left-hand side and the right-hand side values of the inequality respectively, is determined by the ratio $(RHS - LHS)/RHS$. We apply parallel processing with 12 threads of implementation and use default branching rules provided by CPLEX.

## 5. Computational study

We performed a computational study on a set of test problems based on seven VRP instances (available at http://neumann.hec.ca/chairedistributique/data/), namely, *p*01, *p*02, *p*03, *p*04, *p*05, *p*11 and *p*12 in which the number of vertices range between 51 and 200. Note that there are actually 14 instances on this website; however, only seven of them are different with respect to node coordinates and demand values. The remaining seven instances differ from the ones above with respect to the number of vehicles involved and service times, which are not relevant parameters for our problem. The reason for choosing these instances for our tests instead of TSP instances is that we consider the TCMCSP with arbitrary demands rather than unit demands. We experimented with varying values of the parameters $r$, $L$ and $\alpha$ in our tests.

The experiments were conducted on a 64-bit machine with Intel Xeon E5-2630 v2 processor at 2.60 GHz and 96 GB of RAM. All models and algorithms were implemented in Java by invoking CPLEX 12.6 with Concert Technology. The time limit is set to one hour.

**Table 2**
Results with branch-and-cut scheme 1.

| Solution times (s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | | | $r=10$ | | | $r=20$ | | |
| Name | $n$ | $\alpha$ | $L_1$ | $L_2$ | $L_3$ | $L_1$ | $L_2$ | $L_3$ |
| *p*01 | 51 | 0.50 | 4.77 | 1.43 | 0.71 | 1.58 | 0.78 | 0.37 |
| | | 0.75 | 7.16 | 9.32 | 0.58 | 3.33 | 1.10 | 0.44 |
| *p*02 | 76 | 0.50 | 92.24 | 28.29 | 13.06 | 26.79 | 0.82 | 2.43 |
| | | 0.75 | 1236.86 | 19.27 | 3.10 | 121.28 | 2.86 | 4.65 |
| *p*03 | 101 | 0.50 | (3.29) | 76.31 | 3.28 | 360.18 | 21.64 | 7.48 |
| | | 0.75 | (11.31) | 75.81 | 5.07 | 2278.70 | 20.15 | 7.85 |
| *p*04 | 151 | 0.50 | (14.76) | 82.71 | 29.67 | 1275.24 | 49.62 | 22.84 |
| | | 0.75 | (19.06) | 47.65 | 26.08 | 1965.48 | 45.01 | 46.20 |
| *p*05 | 200 | 0.50 | (8.95) | 915.06 | (0.09) | (1.57) | 360.07 | 3046.80 |
| | | 0.75 | (16.78) | 2400.86 | (0.05) | (2.26) | 455.59 | 2238.77 |
| *p*11 | 121 | 0.50 | (117.76) | (28.17) | 43.68 | (70.41) | (20.10) | 176.11 |
| | | 0.75 | (114.68) | (22.81) | 246.03 | (106.07) | (9.60) | 190.07 |
| *p*12 | 101 | 0.50 | (34.41) | 1140.06 | 120.74 | (4.43) | 584.69 | 65.27 |
| | | 0.75 | (60.27) | (2.21) | 232.87 | (9.27) | 2322 | 131.35 |
| Root gaps (%) | | | | | | | | |
| *p*01 | 51 | 0.50 | 17.46 | 3.07 | 0.35 | 6.08 | 0.69 | 0.25 |
| | | 0.75 | 23.92 | 2.71 | 0.38 | 4.57 | 0.71 | 0.20 |
| *p*02 | 76 | 0.50 | 16.01 | 3.07 | 0.41 | 8.23 | 0.69 | 0.27 |
| | | 0.75 | 22.89 | 1.47 | 0.15 | 6.04 | 0.08 | 0.13 |
| *p*03 | 101 | 0.50 | 17.44 | 1.80 | 0.18 | 9.42 | 0.25 | 0.22 |
| | | 0.75 | 19.47 | 1.04 | 0.09 | 5.74 | 0.11 | 0.11 |
| *p*04 | 151 | 0.50 | 18.79 | 2.84 | 0.12 | 5.27 | 2.23 | 0.96 |
| | | 0.75 | 21.23 | 0.30 | 0.47 | 2.36 | 1.01 | 0.47 |
| *p*05 | 200 | 0.50 | 12.87 | 1.65 | 1.56 | 4.55 | 3.43 | 1.30 |
| | | 0.75 | 18.43 | 1.04 | 0.76 | 3.05 | 1.57 | 0.63 |
| *p*11 | 121 | 0.50 | 134.40 | 44.31 | 10.85 | 101.86 | 32.24 | 10.71 |
| | | 0.75 | 126.07 | 28.52 | 5.14 | 116.82 | 14.64 | 5.08 |
| *p*12 | 101 | 0.50 | 89.72 | 30.20 | 11.17 | 53.41 | 30.30 | 14.03 |
| | | 0.75 | 93.87 | 18.25 | 5.20 | 42.84 | 14.62 | 6.94 |

**Table 3**
Results with branch-and-cut scheme 2.

| Solution times (s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | | | $r=10$ | | | $r=20$ | | |
| Name | $n$ | $\alpha$ | $L_1$ | $L_2$ | $L_3$ | $L_1$ | $L_2$ | $L_3$ |
| $p01$ | 51 | 0.50 | 4.65 | 2.76 | 0.63 | 2.52 | 0.74 | 0.86 |
| | | 0.75 | 6.06 | 3.77 | 0.96 | 3.63 | 0.81 | 0.69 |
| $p02$ | 76 | 0.50 | 38.63 | 7.45 | 6.68 | 34.22 | 1.26 | 3.76 |
| | | 0.75 | 1030.43 | 8.26 | 2.46 | 416.02 | 1.73 | 3.14 |
| $p03$ | 101 | 0.50 | 405.66 | 49.01 | 2.33 | 60.21 | 14.83 | 3.01 |
| | | 0.75 | (10.70) | 23.63 | 7.69 | 548.01 | 27.62 | 2.49 |
| $p04$ | 151 | 0.50 | (11.95) | 33.54 | 14.84 | 351.40 | 34.15 | 20.19 |
| | | 0.75 | (17.83) | 31.80 | 68.87 | 718.63 | 21.26 | 31.76 |
| $p05$ | 200 | 0.50 | (5.66) | 682.59 | (0.02) | 973.62 | 118.26 | 1039.73 |
| | | 0.75 | (14.44) | 1253.14 | 1771.94 | 1499.60 | 113.69 | 323.85 |
| $p11$ | 121 | 0.50 | (16.05) | (7.07) | 40.25 | (19.57) | (0.63) | 82.88 |
| | | 0.75 | (55.53) | (4.40) | 109.03 | (69.85) | (0.93) | 101.50 |
| $p12$ | 101 | 0.50 | (4.79) | 98.19 | 14.59 | 132.85 | 72.91 | 19.80 |
| | | 0.75 | (5.29) | (1.91) | 58.05 | 1186.50 | 688.58 | 39.93 |
| Root gaps (%) | | | | | | | | |
| $p01$ | 51 | 0.50 | 19.22 | 2.24 | 0.27 | 5.89 | 0.73 | 0.22 |
| | | 0.75 | 25.56 | 2.45 | 0.27 | 4.74 | 0.38 | 0.11 |
| $p02$ | 76 | 0.50 | 15.75 | 1.24 | 0.27 | 6.09 | 0.10 | 0.21 |
| | | 0.75 | 23.44 | 1.13 | 0.01 | 5.54 | 0.05 | 0.10 |
| $p03$ | 101 | 0.50 | 15.50 | 0.21 | 0.07 | 6.20 | 0.03 | 0.06 |
| | | 0.75 | 19.43 | 0.16 | 0.04 | 4.44 | 0.01 | 0.02 |
| $p04$ | 151 | 0.50 | 15.18 | 0.06 | 0.04 | 3.02 | 0.09 | 0.06 |
| | | 0.75 | 20.31 | 0.07 | 0.06 | 1.49 | 0.04 | 0.03 |
| $p05$ | 200 | 0.50 | 7.90 | 0.33 | 0.32 | 2.01 | 0.18 | 0.16 |
| | | 0.75 | 15.74 | 0.36 | 0.16 | 1.11 | 0.08 | 0.08 |
| $p11$ | 121 | 0.50 | 77.38 | 20.06 | 0.49 | 80.17 | 15.05 | 0.41 |
| | | 0.75 | 101.55 | 9.73 | 0.23 | 102.69 | 6.10 | 0.19 |
| $p12$ | 101 | 0.50 | 72.02 | 14.53 | 4.53 | 38.72 | 12.96 | 4.65 |
| | | 0.75 | 72.68 | 11.34 | 2.09 | 37.79 | 7.40 | 2.18 |

In our computations, we assume that $r_i = r$ for every $i \in V$, and we used 10 and 20 for the coverage distance $r$. These values are about 25% and 50% of the average edge cost, which is approximately 37 when computed over all instances. To choose the time/distance constraint parameter $L$, we first solved the TSP for each instance. Let $z_i^*(TSP)$ denote the cost of an optimal TSP solution for instance $i$. Then, three different values of $L$, namely $L_1$, $L_2$ and $L_3$, were obtained by rounding 25%, 50% and 75% of the number $z_i^*(TSP)$ for instance $i$. Finally, the partial coverage parameter $\alpha$ was set to 50% and 75%. Therefore, we performed our computational study with 12 different parameter configurations for each of the seven test problems, which means that we attempted to solve 84 instances in total.

### 5.1. Comparison of the proposed algorithms

The results of our computational study are provided in Tables 1–5. Each table demonstrates the solution times (in seconds) and root gap values (percentage gap between the final root relaxation bound and the optimal value) for all instances obtained through a different solution scheme. If an instance cannot be solved to optimality within the time limit, we report the final gap (percentage gap between the objective function value of the best integer solution and the best upper bound) in parenthesis.

We tested the flow-based formulation on a total of 48 instances containing up to 100 customers, namely with $p01$, $p02$, $p03$ and $p12$. The results are provided in Table 1.

Next, we tested our initial branch-and-cut method, where only the connectivity constraints are separated at integer points of the solution tree. The results in Table 2 show that 61 of the 84 instances can be solved optimally within one hour and the average solution time for these instances is 372.13 s. Regarding the remaining 23 instances, the final gap is 29.49% on average. Additionally, the root gap values indicate that the upper bounds given by the relaxation (1)–(5), (7)–(10) are quite weak. In particular, the average and worst gaps are 15.75% and 134.40%, respectively, for the initial branch-and-cut scheme.

When we compare the results in Tables 1 and 2, we observe the following: with the flow formulation, 19 out of 42 instances cannot be solved to optimality within one hour while 12 of these 19 instances can be solved optimally by our first branch-and-cut scheme. Regarding the remaining 7 instances, the final gaps of our first branch-and-cut algorithm are 51.5% less (on average) than that of the flow formulation. Finally, the instances for which an optimal solution can be found by the flow formulation require 98% less solution time on average when solved by our first branch-and-cut algorithm. Hence, we can conclude that the first branch-and-cut algorithm outperforms the flow formulation.

Next, the test instances are solved with our branch-and-cut scheme in which connectivity constraints are separated not only at the integer points, but also at the fractional points of the solution tree. Based on some preliminary experiments, separation for fractional solutions is performed only at the root node of the

**Table 4**
Results with branch-and-cut scheme 3.

| Solution times (s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | | | $r=10$ | | | $r=20$ | | |
| Name | $n$ | $\alpha$ | $L_1$ | $L_2$ | $L_3$ | $L_1$ | $L_2$ | $L_3$ |
| *p*01 | 51 | 0.50 | 1.33 | 0.54 | 0.56 | 0.37 | 0.53 | 0.74 |
| | | 0.75 | 1.03 | 0.60 | 0.70 | 0.17 | 0.10 | 0.94 |
| *p*02 | 76 | 0.50 | 2.01 | 3.17 | 3.75 | 0.48 | 1.47 | 2.93 |
| | | 0.75 | 4.65 | 1.51 | 0.63 | 2.57 | 1.40 | 3.23 |
| *p*03 | 101 | 0.50 | 5.78 | 5.74 | 3.22 | 6.23 | 2.55 | 1.88 |
| | | 0.75 | 8.76 | 6.74 | 5.48 | 5.98 | 3.08 | 2.09 |
| *p*04 | 151 | 0.50 | 30.93 | 15.96 | 16.30 | 13.88 | 29.92 | 26.70 |
| | | 0.75 | 24.19 | 3.76 | 30.18 | 19.11 | 23.72 | 39.55 |
| *p*05 | 200 | 0.50 | 93.32 | 160.44 | 341.83 | 70.62 | 99.51 | 662.62 |
| | | 0.75 | 139.33 | 236.39 | 211.76 | 41.50 | 144.36 | 270.73 |
| *p*11 | 121 | 0.50 | 38.73 | 50.56 | 8.51 | 165.79 | 118.82 | 11.13 |
| | | 0.75 | 260.89 | 84.13 | 9.13 | 1043.97 | 24 | 12.49 |
| *p*12 | 101 | 0.50 | 9.12 | 4.85 | 9.89 | 5.11 | 6.54 | 16.23 |
| | | 0.75 | 11.07 | 22.09 | 12.63 | 13.33 | 10.15 | 8.33 |
| Root gaps (%) | | | | | | | | |
| *p*01 | 51 | 0.50 | 0.59 | 0.17 | 0.22 | 0.97 | 0.05 | 0.22 |
| | | 0.75 | 0.67 | 0.01 | 0.23 | 0.05 | 0 | 0.11 |
| *p*02 | 76 | 0.50 | 1.30 | 0.22 | 0.11 | 0 | 0.06 | 0.21 |
| | | 0.75 | 1.42 | 0.01 | 0.01 | 0.69 | 0.02 | 0.10 |
| *p*03 | 101 | 0.50 | 0.07 | 0.14 | 0.07 | 0.08 | 0 | 0.06 |
| | | 0.75 | 0.97 | 0.02 | 0.04 | 0.43 | 0 | 0.03 |
| *p*04 | 151 | 0.50 | 0.52 | 0.05 | 0.02 | 0 | 0.07 | 0.05 |
| | | 0.75 | 0.02 | 0 | 0.06 | 0 | 0.03 | 0.02 |
| *p*05 | 200 | 0.50 | 0.04 | 0.11 | 0.26 | 0.07 | 0.11 | 0.16 |
| | | 0.75 | 0.19 | 0.10 | 0.13 | 0 | 0.05 | 0.08 |
| *p*11 | 121 | 0.50 | 10.89 | 3.52 | 0.06 | 13 | 1.48 | 0.15 |
| | | 0.75 | 20.33 | 0.24 | 0.03 | 29.90 | 0.03 | 0.07 |
| *p*12 | 101 | 0.50 | 0.98 | 0.34 | 0.20 | 0.99 | 0.44 | 0.32 |
| | | 0.75 | 0.71 | 1.35 | 0.11 | 2.14 | 0 | 0.15 |

**Table 5**
Results with branch-and-cut scheme 4.

| Solution times (s) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Instance | | | $r=10$ | | | $r=20$ | | |
| Name | $n$ | $\alpha$ | $L_1$ | $L_2$ | $L_3$ | $L_1$ | $L_2$ | $L_3$ |
| *p*01 | 51 | 0.50 | 1.66 | 0.52 | 0.54 | 0.67 | 0.45 | 0.67 |
| | | 0.75 | 0.98 | 0.61 | 0.76 | 0.26 | 0.09 | 0.78 |
| *p*02 | 76 | 0.50 | 2.78 | 3.08 | 3.75 | 0.52 | 1.39 | 2.80 |
| | | 0.75 | 4.63 | 1.61 | 0.61 | 2.08 | 1.45 | 3.06 |
| *p*03 | 101 | 0.50 | 6.87 | 5.92 | 3.02 | 8.15 | 2.49 | 1.78 |
| | | 0.75 | 9.72 | 6.55 | 5.58 | 6.69 | 3.07 | 2.04 |
| *p*04 | 151 | 0.50 | 45.22 | 15.56 | 15.58 | 12.24 | 30.25 | 26.27 |
| | | 0.75 | 43.74 | 3.85 | 30.78 | 18.37 | 23.68 | 40.51 |
| *p*05 | 200 | 0.50 | 93.24 | 162.44 | 343.41 | 70.75 | 99.10 | 657.46 |
| | | 0.75 | 139.98 | 233.22 | 214.95 | 40.75 | 144.74 | 263.60 |
| *p*11 | 121 | 0.50 | 3.12 | 42.25 | 8.08 | 4.64 | 101.34 | 10.82 |
| | | 0.75 | 8.22 | 53.28 | 9.13 | 9.55 | 43.34 | 12.01 |
| *p*12 | 101 | 0.50 | 7.72 | 4.82 | 9.47 | 9.57 | 6.22 | 15.75 |
| | | 0.75 | 12.87 | 22.13 | 12.34 | 10.16 | 10.23 | 8.09 |
| Root gaps (%) | | | | | | | | |
| *p*01 | 51 | 0.50 | 0.42 | 0.17 | 0.22 | 0.72 | 0.05 | 0.22 |
| | | 0.75 | 0.82 | 0.01 | 0.23 | 0.16 | 0 | 0.11 |
| *p*02 | 76 | 0.50 | 1.29 | 0.22 | 0.11 | 0 | 0.06 | 0.21 |
| | | 0.75 | 1.50 | 0.01 | 0.01 | 0.69 | 0.02 | 0.10 |
| *p*03 | 101 | 0.50 | 0.16 | 0.14 | 0.07 | 0.08 | 0 | 0.06 |
| | | 0.75 | 0.86 | 0.02 | 0.04 | 0.45 | 0 | 0.03 |
| *p*04 | 151 | 0.50 | 0.52 | 0.05 | 0.02 | 0.03 | 0.07 | 0.05 |
| | | 0.75 | 0.01 | 0 | 0.06 | 0.01 | 0.03 | 0.02 |
| *p*05 | 200 | 0.50 | 0.04 | 0.11 | 0.26 | 0.07 | 0.11 | 0.16 |
| | | 0.75 | 0.19 | 0.10 | 0.13 | 0 | 0.05 | 0.08 |
| *p*11 | 121 | 0.50 | 0.18 | 3.52 | 0.06 | 0.03 | 1.47 | 0.15 |
| | | 0.75 | 0.31 | 0.08 | 0.03 | 5.06 | 0.04 | 0.07 |
| *p*12 | 101 | 0.50 | 0.98 | 0.34 | 0.20 | 0.86 | 0.44 | 0.32 |
| | | 0.75 | 0.56 | 1.35 | 0.11 | 2.22 | 0 | 0.15 |

branch-and-cut tree as mentioned in the previous section. Note that this is also the case for the remaining two schemes. Table 3 reports the results obtained with our second branch-and-cut algorithm. The number of instances for which an optimal solution is found within one hour is 67 with an average solution time of 216 s, and the average final gap is 14.5% for the remaining 17 instances. Moreover, we can observe that introducing some violated connectivity constraints at the root node of the solution tree strengthens the linear relaxation bounds of (1)–(5), (7)–(10) and reduces the average and maximum root gap values to 11.02% and 102.69%, respectively. Compared to the initial scheme, the solution times (or the optimality gaps for the instances that cannot be solved optimally within the time limit) and the root gaps improve in almost all of the instances.

In our third scheme, we add SLCIs to the initial relaxation (1)–(5), (7)–(10) for $i$ with $0 \notin N_i$ and separate LCIs in addition to connectivity constraints. Based on the results in Table 4, all instances can be solved to optimality within at most 1043.97 s. The average solution time is 57.11 s, indicating a 74% decrease compared to the previous scheme. There is also a significant improvement regarding the root gaps. In particular, the average root gap decreases to 1.18%, which implies a 90% reduction with respect to our second scheme. Overall, introducing SLCIs and violated LCIs during the solution procedure leads to a remarkable improvement both in terms of the solution times and the root relaxation bounds.

Finally, we test the effect of using cover inequalities. In addition to LCIs and connectivity constraints, we also separate cover inequalities in our last branch-and-cut scheme. Furthermore, for every pair of vertices $i, j \in V \setminus \{0\}$ with $c_{\{0,i\}} + c_{\{i,j\}} + c_{\{j,0\}} > L$, we include the inequalities $y_i + y_j \le 1$ and set $x_{ij} = 0$ in the initial relaxation besides adding SLCIs. Although the average solution time decreases to 39.31 s and the average root gap is as low as 0.35% in the final scheme, these reductions are mostly caused by the instance $p11$. In other words, there is no considerable change in terms of the solution times and the root gaps compared to the previous scheme except for a few instances based on $p11$. No violated inequalities of this family are detected for most of the instances. However, we observe that it may be possible to obtain significant improvements in some cases. As an example, the instance $p11$, $\alpha = 0.75$, $r=20$ and $L = 0.25z^*_{p11}(TSP)$ can be solved within 9.55 s with the help of cover inequalities, while the solution time for the same instance was 1043.97 s in our previous scheme.

As indicated above, remarkable improvements can be achieved regarding solution times and linear relaxation bounds with the introduction of LCIs and cover inequalities. Based on the results of our worst and best branch-and-cut algorithms, decrease in the average solution time is about 90% and the average root gap reduces by 98%. In our best algorithm (the last scheme), the maximum solution time is 657.46 s while there are 23 instances that cannot be solved optimally within the time limit of one hour by our initial algorithm. This shows the effectiveness of LCIs and cover inequalities in terms of speeding up the solution procedure.

Moreover, the largest root gap with our initial algorithm is 134.40%, whereas it is only 5.06% in the last one, which is another evidence of the power of these inequalities in strengthening the linear relaxation bounds of our formulation.

### *5.2. Impacts of parameter changes on the optimal solutions*

In a network optimization problem, topology of the underlying graph and the adopted parameter setting are among the key determinants of the optimal solution structure. In order to understand the effects of these factors, we investigate the structure of the optimal TCMCSP solutions by evaluating the percentage of demand covered, the number of tour stops and partially covered vertices as well as tour and coverage diameters. The diameter of a tour is defined as the largest distance between two vertices on the tour. Similarly, the coverage diameter is the largest distance between any pair of vertices that are fully or partially covered. Note that the vertices are scattered throughout the graph for $p01$, $p02$, $p03$, $p04$, $p05$, while they are located in several clusters for $p11$ and $p12$. In all the figures below, the filled circles represent tour stops and partially covered vertices while the empty ones correspond to isolated vertices.

As expected, the percentage of total demand covered increases with larger values of the partial coverage parameter $\alpha$. The tour diameter tends to increase as well for the instances where the underlying graph is scattered. This is mostly achieved by visiting fewer vertices in an attempt to cover more vertices partially. The increase in the tour diameter and in the total demand covered obtained as a result of using a greater $\alpha$ value is most notable when the value of $L$; i.e., the maximum tour length parameter, is small, because the coverage diameter is usually large enough to accommodate almost all vertices of the graph sufficiently close to the tour when $L$ is large. Fig. 1 shows an example for $\alpha = 0.50$ and $\alpha = 0.75$ when $L$ and the coverage distance $r$ are fixed. For the instances where the underlying graph is clustered, changing the value of $\alpha$ does not usually affect the tour diameter, because the coverage cannot be enhanced unless the tour is expanded to reach an unvisited cluster, which is not possible without increasing $L$ sufficiently. An example is depicted in Fig. 2.

Regarding the impact of tour length restriction on the optimal solution structure, we observed that a higher percentage of the total demand can be covered and the coverage diameter expands as $L$ gets larger. In scattered graphs, more vertices are visited while fewer vertices are partially covered when $L$ is increased. Moreover, changes in the percentage of demand that is covered on-tour and off-tour have a similar impact when compared to changes in the number of tour stops and partially covered vertices, respectively. An illustration of the impact of maximum tour length is given in Fig. 3 for a scattered graph.

On the other hand, in clustered graphs, the total demand covered is improved first by attempting to reach as many clusters as possible, while respecting the tour length constraint imposed by $L$, and then by visiting the maximum possible number of vertices in these clusters. Therefore, even though the number of tour stops does not necessarily get larger as $L$ increases, the percentage of demand that is fully covered becomes higher according to our

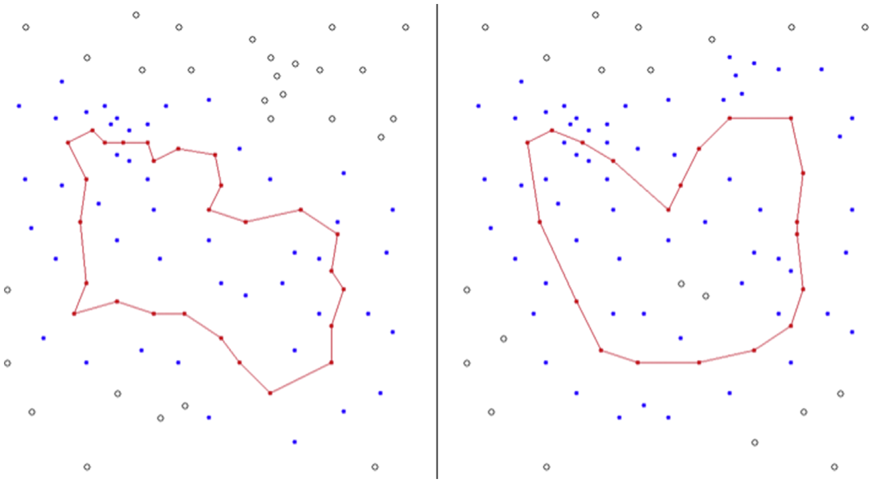

**Fig. 1.** The optimal solutions of $p03$ with $L = L_1$, $r$=10 for $\alpha = 0.5$ and $\alpha = 0.75$, respectively.

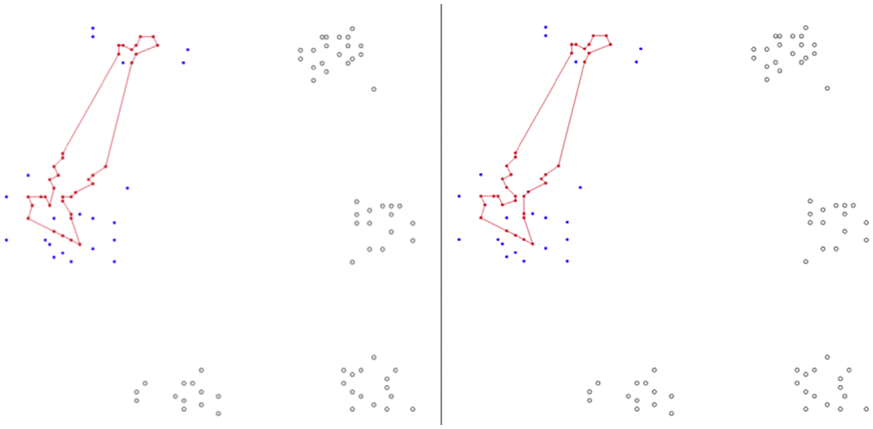

**Fig. 2.** The optimal solutions of $p11$ with $L = L_1$, $r$=10 for $\alpha = 0.5$ and $\alpha = 0.75$, respectively.

observations. Furthermore, if the increase in $L$ is enough to reach some previously unvisited clusters; i.e., if the growth in the tour diameter is significant, then the percentage of partially covered demand becomes larger. When $L$ increases further (but not enough to visit another cluster), more vertices in the visited clusters are included in the tour, resulting in a decrease in the percentage of partially covered demand. An example of this situation is illustrated in Fig. 4.

Finally, we explored how the optimal solutions are affected by the changes in coverage distance. Raising the value of the coverage distance parameter increases the percentage of total demand covered and the coverage diameter in general. Note that this increase is more significant in scattered graphs, whereas the optimal solution may only slightly change or even remain unchanged in clustered graphs unless the increase in $r$ is sufficient to cover some vertices without visiting the clusters they belong to. Based on our results, we observed that increasing $r$ usually gives rise to a decrease in the tour diameter. This enables visiting more vertices while maintaining the same coverage level especially when the underlying graph is scattered and dense. Fig. 5 exhibits how the optimal solution changes depending on the coverage distance for fixed $\alpha$ and $L$. However, regarding the clustered instances where $r$ is increased enough to allow some vertices in a different cluster to be covered by the tour stop(s), we noticed that the tour diameter increases and fewer vertices are visited. Such an example is depicted in Fig. 6.

Overall, we can conclude that the maximum tour length $L$ is the most critical parameter that determines the optimal solutions based on the test instances used in our experiments. The impact of changing $L$ can be remarkable especially when the underlying problem graph is clustered. Even though the parameters $\alpha$ and $r$ affect the optimal solution in terms of which vertices should be designated as tour stops and which ones should be partially covered in order to ensure maximum coverage, the improvement achieved by increasing the values of these parameters cannot go beyond a certain limit (assuming that $\alpha < 1$) under the same tour length restriction. This is actually an intuitive result since with sufficiently large $L$ values, all vertices would become tour stops. However, in general we can remark that as $\alpha$ gets larger, the optimal solution tends to contain more partially covered vertices and fewer tour stops scattered through a slightly wider area (the tour diameter is larger). Conversely, increasing $r$ often leads to a reduction in the tour diameter, which facilitates visiting more vertices, except for some clustered graphs. Finally, it is clear that the impact of changing the value of a parameter on the optimal solutions also depends on the underlying problem graph.

## 6. Conclusion

We have considered the time constrained maximal covering salesman problem (TCMCSP) in which the goal is to find a tour visiting a subset of the vertices that maximizes the amount of demand covered subject to an upper bound on the tour length. This problem is practically relevant in cases where it is not efficient to visit every demand point separately. Integrating the notion of coverage into a routing scheme; i.e., satisfying the demand of multiple customers through each customer on the route, may provide means to increase system efficiency by utilizing the available resources more effectively. This paper presents an efficient solution approach to a problem that unifies coverage and routing.

In the TCMCSP, we assumed that the demand of a vertex is fully

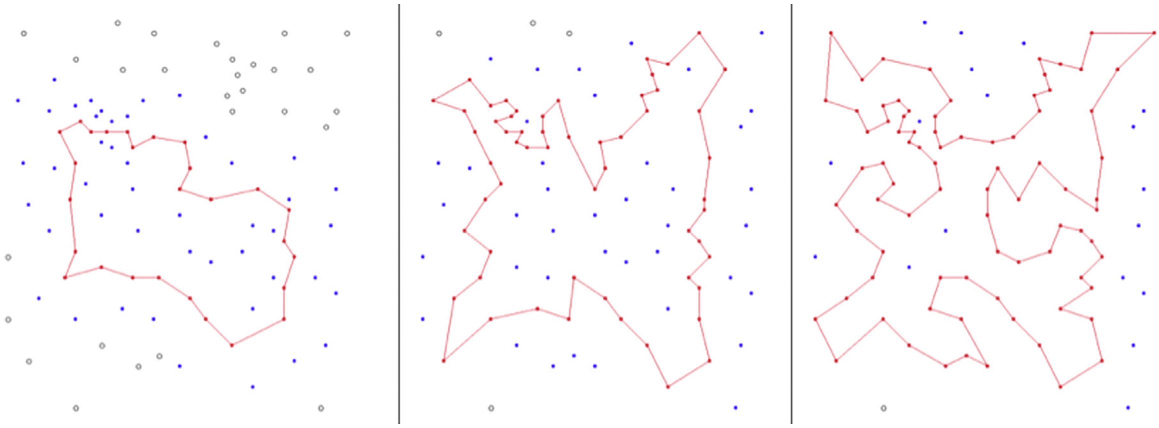

**Fig. 3.** The optimal solutions of $p03$ with $\alpha = 0.5$, $r=10$ for $L = L_1$, $L = L_2$ and $L = L_3$, respectively.

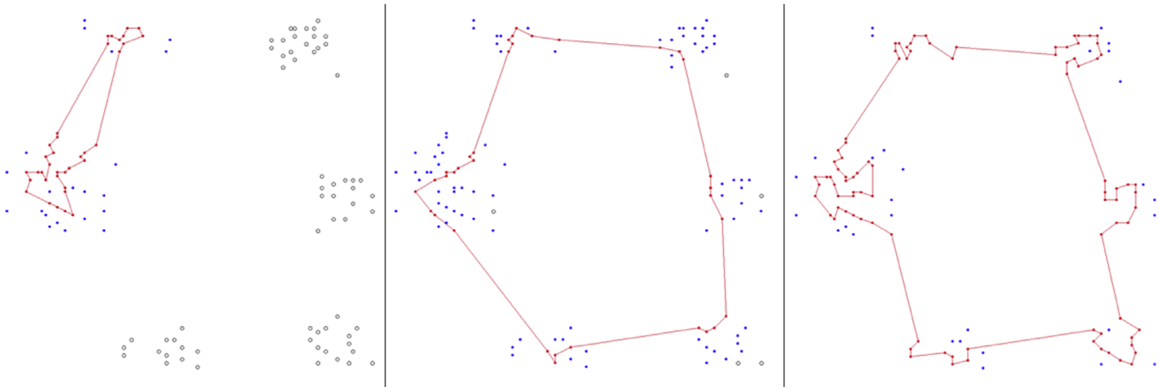

**Fig. 4.** The optimal solutions of $p11$ with $\alpha = 0.5$, $r=10$ for $L = L_1$, $L = L_2$ and $L = L_3$, respectively.

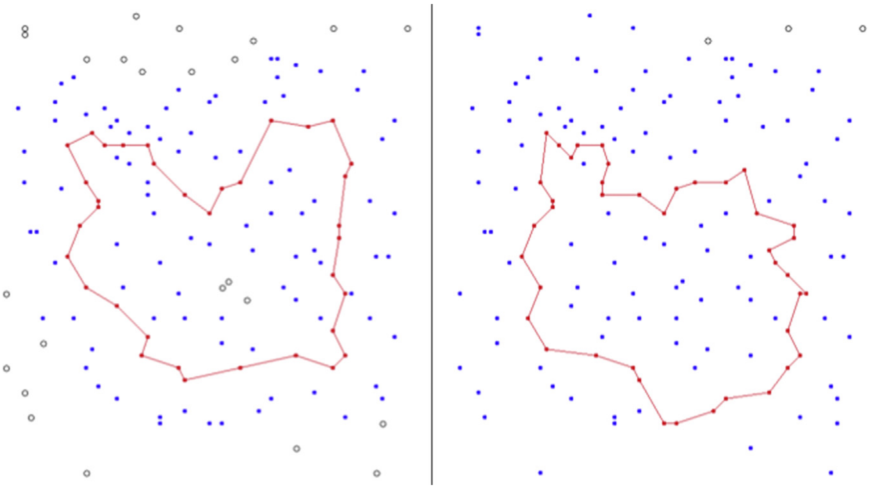

**Fig. 5.** The optimal solutions of $p04$ with $\alpha = 0.5$, $L = L_1$ for $r$=10 and $r$=20, respectively.

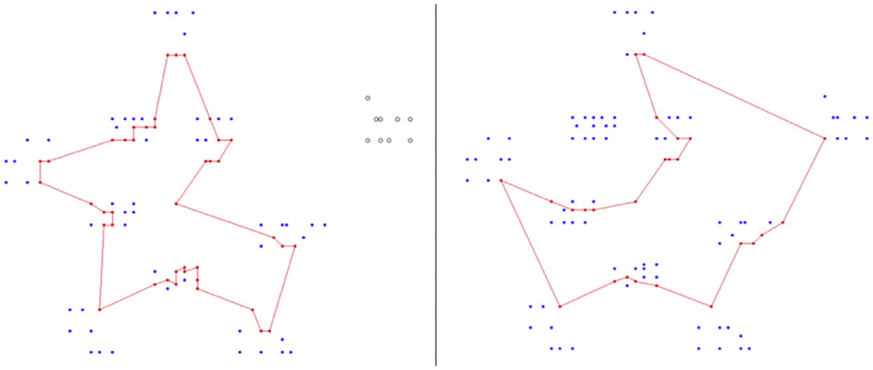

**Fig. 6.** The optimal solutions of $p12$ with $\alpha = 0.75$, $L = L_2$ for $r$=10 and $r$=20, respectively.

covered if it is visited, partially covered if it is not visited but sufficiently close to some vertex on the tour, and not covered otherwise. We modeled the problem on an undirected network and since our formulation involves exponentially many connectivity constraints, we proposed branch-and-cut algorithms to solve the TCMCSP. We presented simple optimality cuts and two families of valid inequalities, namely the lifted connectivity inequalities (LCIs) and cover inequalities. Four branch-and-cut schemes were devised to evaluate the impact of LCIs and cover inequalities on solution times and LP relaxation bounds of our formulation.

We also adapted the flow formulation presented in [1] to model the problem. Computational experiments demonstrated the superiority of using a branch-and-cut solution approach over the flow formulation. Moreover, the results indicated the effectiveness of LCIs and cover inequalities, both of which were shown to be quite powerful in increasing the quality of linear relaxation bounds of our formulation and accelerating the solution procedure.

Finally, we investigated the effects of the partial coverage multiplier ($\alpha$), maximum tour length ($L$) and coverage distance ($r$) parameters on the optimal solution structure. We conclude that $L$ has the largest impact on the optimal solutions. Additionally, there is a tendency towards visiting fewer vertices as $\alpha$ increases, while extending $r$ usually results in the reverse behavior.

## Acknowledgments

We thank the area editor and two anonymous referees for their helpful comments on a previous version of this paper. The research of the first author is supported by the Scientific and Technological Research Council of Turkey (TUBITAK), Grant Number: BIDEB-2211-A.

## References

[1] Naji-Azimi Z, Salari M. The time constrained maximal covering salesman problem. Appl Math Model 2014;38(15):3945–57.

[2] Current JR, Schilling DA. The covering salesman problem. Transp Sci 1989;23 (3):208–13.

[3] Arkin EM, Hassin R. Approximation algorithms for the geometric covering salesman problem. Discret Appl Math 1994;55(3):197–218.

[4] Current JR, Schilling DA. The median tour and maximal covering tour problems: formulations and heuristics. Eur J Oper Res 1994;73(1):114–26.

[5] Golden BL, Naji-Azimi Z, Raghavan S, Salari M, Toth P. The generalized covering salesman problem. INFORMS J Comput 2012;24(4):534–53.

[6] Salari M, Reihaneh M, Sabbagh MS. Combining ant colony optimization algorithm and dynamic programming technique for solving the covering salesman problem. Comput Ind Eng 2015;83:244–51.

[7] Shaelaie MH, Salari M, Naji-Azimi Z. The generalized covering traveling salesman problem. Appl Soft Comput 2014;24:867–78.

[8] Gendreau M, Laporte G, Semet F. The covering tour problem. Oper Res 1997;45 (4):568–76.

[9] Hodgson MJ, Laporte G, Semet F. A covering tour model for planning mobile health care facilities in suhum district, Ghana. J Reg Sci 1998;38(4):621–38.

[10] Motta L, Ochi LS, Martinhon C. Grasp metaheuristics for the generalized covering tour problem. In: MIC2001-4th metaheuristics international conference. Porto, Portugal: Citeseer; 2001.

[11] Baldacci R, Boschetti MA, Maniezzo V, Zamboni M. Scatter search methods for the covering tour problem. In: Metaheuristic optimization via memory and evolution. New York: Springer; 2005. p. 59–91.

[12] Kubik P. Heuristic solution approaches for the covering tour problem [Ph.D. thesis]. Universität Wien; 2007.

[13] Hachicha M, Hodgson MJ, Laporte G, Semet F. Heuristics for the multi-vehicle covering tour problem. Comput Oper Res 2000;27(1):29–42.

[14] Naji-Azimi Z, Renaud J, Ruiz A, Salari M. A covering tour approach to the location of satellite distribution centers to supply humanitarian aid. Eur J Oper Res 2012;222(3):596–605.

[15] Oliveira WA, Mello MP, Moretti AC, Reis EF. The multi-vehicle covering tour problem: building routes for urban patrolling. arXiv:1309.5502.

[16] Jozefowiez N, Semet F, Talbi EG. The bi-objective covering tour problem. Comput Oper Res 2007;34(7):1929–42.

[17] Nolz PC, Doerner KF, Gutjahr WJ, Hartl RF. A bi-objective metaheuristic for disaster relief operation planning. In: Advances in multi-objective nature inspired computing. Berlin: Springer; 2010. p. 167–87.

[18] Tricoire F, Graf A, Gutjahr WJ. The bi-objective stochastic covering tour problem. Comput Oper Res 2012;39(7):1582–92.

[19] Feillet D, Dejax P, Gendreau M. Traveling salesman problems with profits. Transp Sci 2005;39(2):188–205.

[20] Golden BL, Levy L, Vohra R. The orienteering problem. Nav Res Logist 1987;34 (3):307–18.

[21] Laporte G, Martello S. The selective travelling salesman problem. Discret Appl Math 1990;26(2–3):193–207.

[22] Kataoka S, Morito S. An algorithm for single constraint maximum collection problem. J Oper Res Soc Jpn 1988;31(4):515–30.

[23] Vansteenwegen P, Souffriau W, Van Oudheusden D. The orienteering problem: a survey. Eur J Oper Res 2011;209(1):1–10.

[24] Fouilhoux P, Karasan OE, Mahjoub AR, Özkök O, Yaman H. Survivability in hierarchical telecommunications networks. Networks 2012;59(1):37–58.

[25] Stoer M, Wagner F. A simple min-cut algorithm. J Assoc Comput Mach 1997;44 (4):585–91.

[26] Edmonds J, Karp RM. Theoretical improvements in algorithmic efficiency for network flow problems. J Assoc Comput Mach 1972;19(2):248–64.