



Time-constrained maximal covering routing problem

Afsaneh Amiri¹ · Majid Salari¹

Received: 6 January 2018 / Accepted: 21 November 2018 / Published online: 4 December 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

We introduce the time-constrained maximal covering routing problem (TCMCRP), as a generalization of the covering salesman problem. In this problem, we are given a central depot, a set of facilities and several customers which are located within a pre-determined coverage distance of available facilities. Each facility can supply the demand of some customers which are within its coverage radius. Starting from the depot, the goal is to maximize the total number of covered customers, by constructing a set of p length constraint Hamiltonian cycles. We have proposed a mixed integer linear programming model and three heuristic algorithms, namely iterated local search (ILS), tabu search (TS) and variable neighborhood search (VNS), to solve the problem. Extensive computational tests on this problem and some of its variants clearly indicate the effectiveness of the developed solution methods.

Keywords Covering salesman problem · Transportation · Covering · Mixed integer linear programming · Heuristics

1 Introduction and literature review

Despite remarkable advances in healthcare services, we are still witnessing the emergence and spreading of several new diseases around the world. According to the World Health Organization reports, the most prevalent diseases in 2017 were avian influenza, coronavirus and yellow fever, which have not yet been controlled and eradicated after more than a year (<http://www.who.int/csr/don/archive/year/2018/en/>). Several other reports in recent years (<https://ecdc.europa.eu/en/threats-and-outbreaks>; <https://www.cdc.gov/outbreaks/index.html>) indicate that locating the delivery healthcare teams and field hospitals in the potential areas must be in a way that the maximum number of

✉ Majid Salari
msalari@um.ac.ir

Afsaneh Amiri
Afsaneh_amb@yahoo.com

¹ Department of Industrial Engineering, Ferdowsi University of Mashhad, Mashhad, Iran

patients receive medical treatments in the shortest possible time in order to prevent the outbreaks of disease. Therefore, the problems in which researchers are seeking optimal methods to reduce the traveling time or distance as well as increase the number of satisfied customers are considered as effective ways to deal with the spreading of epidemics. For this reason, in this paper, we discuss a problem which simultaneously increases the number of the covered customers and decreases the length of the traveling distance. Indeed, we have combined the routing and the covering problems in order to benefit from the characteristics of both problems. Several combinatorial optimization problems, in the field of routing and covering, are proposed and developed. In particular, we can refer to the vehicle routing, the maximal covering and the traveling salesman problems. In the following, the mentioned problems and their variations are discussed first and then we outline the proposed problem, its characteristics and applications.

The vehicle routing problem (VRP) is a combinatorial problem, seeking to visit a number of customers by utilizing a set of vehicles which are located at a central depot (Dantzig and Ramser 1959). Most popular variants of this problem are the VRP with time windows (VRPTW), the capacitated VRP (CVRP) and the multi-depot VRP (MDVRP). In the VRPTW, each customer must be visited within a certain time interval. The problem has several applications including the meal delivery, the waste collection and the school bus routing problems (Jabali et al. 2015; Bae and Moon 2016; Buhrkal et al. 2012; Hiermann et al. 2016). The vehicles are capacitated in the CVRP, and several applications, namely road transportation and network routing problems, have been introduced for this problem (Faulin et al. 2011; Sørensen and Schittekat 2013; Hosseini-abadi et al. 2017). The MDVRP uses multiple depots instead of one depot to construct the routes. Delivery of food and drugs, chemical and oil products is some applications of the MDVRP (Wasner and Zäpfel 2004; Kergosien et al. 2013; Lalla-Ruiz et al. 2016). For further studies, readers can refer to Toth and Vigo (2001) and Golden et al. (2008).

The maximal covering problem (MCP) was introduced by Church and Velle (1974). The goal of the MCP is to maximize the amount of demand that can be covered by a limited number of facilities within a specified distance or time. Generally, in the MCP we are given several customers and facilities; each facility can respond some customers' demand which is located within its pre-determined distance. The goal is maximizing the number of covered customers by using a limited number of facilities. In Pirkul and Schilling (1991) the capacitated MCP was introduced in which each facility has a limited capacity.

The covering salesman problem (CSP) is an extension of the traveling salesman problem (TSP) which is seeking construction of a minimum length tour over a subset of n nodes such that each unvisited node is within a pre-determined covering distance of at least one of the visited nodes. Routing of rural healthcare delivery teams, locating aircraft for overnight delivery systems and locating mail boxes are some applications of the CSP (Current and Schilling 1989). Gendreau et al. (1997) introduced the covering tour problem in which the vertices are composed of three sets including a set of vertices that must be visited (S_1), a set of vertices that can be visited (S_2), and a set of vertices that must be covered (S_3) by visited ones. The purpose is to find a minimum length cycle over the vertices of S_1 and a subset of those belonging to S_2 in order to cover all the vertices of S_3 . One of the applications is locating the post boxes among potential sites in a way that all of the users located in a reasonable distance from boxes are

covered and the cost of collection tour over all selected boxes is minimized. In Golden et al. (2012) the authors provided a generalized version of the CSP. The objective is to find a minimum cost route over a subset of vertices such that every unrouted vertex has to be covered at least k times by those vertices that are visited on the tour. As an application, in routing of rural healthcare delivery teams, in each day, a limited number of people could be served by the team; therefore, some points need to be visited more than once. Another related paper is Current and Schilling (1994) in which two bi-objective problems known as the median tour problem (MTP) and the maximal covering tour problem (MCTP) are introduced. In each problem, the vehicle must visit only p (out of n) nodes on the network. Furthermore, minimization of the total tour length is considered as one of the objectives of both problems. The second objective is maximizing the access of the tours to the nodes that are not directly on the tour. This objective in the MTP is realized by minimizing the total distance traveled from nodes that are not on the tour to the nearest stop on the tour, while in the MCTP it is achieved by minimizing the demand of nodes which are not covered by the stops on the tour. These two problems have numerous real-world applications such as rural healthcare delivery, designing of distributed computer networks and design of hierarchical transportation networks (Current and Schilling 1994).

The other related problem is the orienteering problem (OP) which is also known as the maximum collection problem and was first introduced in Tsiligirides (1984). This problem is a development of the TSP and also known as the selective TSP. Here, each vertex has an associated profit and instead of visiting all of the vertices, a subset of vertices with the maximum profit is visited, while the tour length is not allowed to exceed from a specified threshold (Laport and Martello 1990). The team orienteering problem (TOP) is an extension of the previous problem and consists of determining m length constraint circuits. This problem has been proposed in Butt and Cavalier (1994) with the name multiple tour maximum collection problem (MTMCP). Several variations of the TOP are presented in the literature including TOP with time windows (Vansteenwegen et al. 2009; Gunawan et al. 2015; Souffriau et al. 2013) and the capacitated orienteering problem in which vehicles have limited capacities (Archetti et al. 2010; Aras et al. 2011; Zachariadis and Kiranoudis 2011; Bock and Sanità 2015). For a comprehensive survey of OP and TOP, the reader is referred to Vansteenwegen et al. (2011) and Gunawan et al. (2016).

In another relevant problem which is referred to as the ring star problem, the goal is to minimize the length of a cycle which is constructed over a subset of vertices, as well as to minimize the allocation cost which is imposed to the problem by the remaining vertices that are not located in the cycle and served by the closest vertex in the cycle. This problem is an extension of the location allocation problem and has been introduced firstly by Lee et al. (1996) and developed and solved in Labbé et al. (2004, Baldacci et al. (2007), and Sundar and Rathinam (2017).

In this paper, we present the time-constrained maximal covering routing problem (TCMCRP) which is a generalization of the time-constrained maximal covering salesman problem (TCMCSP) (Naji-Azimi and Salari 2014). In this paper, we develop the TCMCSP by considering multiple vehicles instead of just a single vehicle due to the fact that in real world, we have to deal with the problems with several vehicles. In the TCMCRP, we are given a set of vertices including a central depot, a set of customers

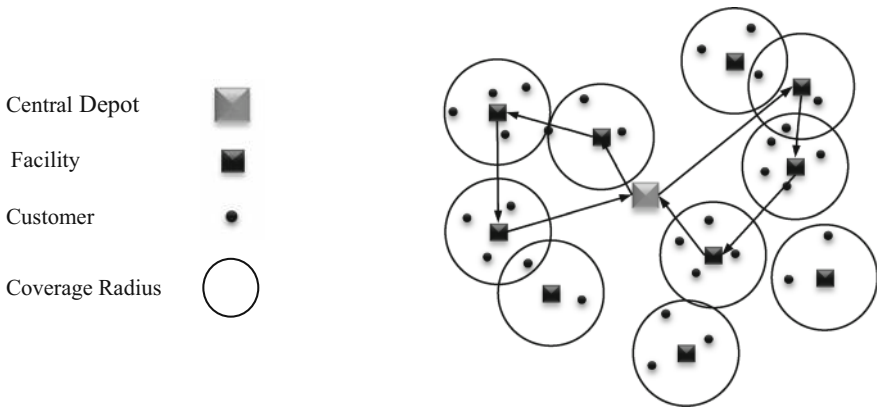


Fig. 1 An illustrative example of the TCMCRP

and a set of facilities. Each facility can supply the demand of some customers within its pre-specified coverage distance. The goal is maximizing the total number of covered customers by constructing a limited number of length constraint Hamiltonian cycles over a subset of facilities.

Since the TCMCRP is the combination of the CSP and the TOP, it can be applied to all applications already proposed for these two problems. For example, in rural healthcare delivery system, teams should travel to numerous areas in a limited time to provide general care services like vaccination, emergency services or such activities for prevention of epidemics. Having a limited amount of resources, the teams have to select some areas for visiting in order to maximize the number of individuals that will be covered. Another application is discussed when natural or human disasters like earthquakes, wars or typhoons occur. In these situations, the problem includes the assignment of some locations for establishing field hospitals among the potential areas. The locations should be chosen in a way that the number of injured people who can receive emergency assistance is maximized and the traveled distance from main centers to these locations is minimized. In above examples, maximizing the covered demand subject to the availability of limited resources is the main issue which is the basic characteristic of the TCMCRP model.

An illustrative example is shown in Fig. 1. In this figure, we are given 41 nodes including 10 facilities, 30 customers and a central depot. In this example, 6 facilities are visited by two available vehicles and totally 22 customers have been covered.

The remainder of this paper is organized as follows: Formal description of the problem is provided in Sect. 2. Section 3 describes the characteristics of the heuristic algorithms. Computational results are presented in Sect. 4, followed by conclusions which are reported in Sect. 5.

2 Problem formulation

In this section we develop a flow-based model to formulate the proposed TCMCRP. Flow-based models have been developed in different articles (Dell'Amico et al.

2006; Montané and Galvao 2006). In our developed model, the sub-tour elimination constraints are a generalization of those proposed in Karaoglan et al. (2009) and Kara (2011) for the single-vehicle version of the problem. In particular, the proposed model is a generalization of the already proposed model in Naji-Azimi and Salari (2014) for the TCMCSP.

This TCMCRP is defined on a directed graph $G = (V, A)$ where $V = \{0\} \cup T \cup W$ is the set of vertices. $\{0\}$ represents the central depot at which a set of p homogeneous vehicles $P = \{1, 2, \dots, p\}$ are located. W is the set of customers, and T is the set of facilities. $A = \{(i, j) | i, j \in T \cup \{0\}\}$ is the arc set where each arc is associated with a weight t_{ij} (travel distance or time). The maximum travel length of each vehicle $k \in P$ cannot exceed the given limit L_k .

Covering matrix $D = [d_{ij}]$ is given in which $d_{uv} = 1$ when customer u is located within a pre-specified coverage radius of facility v ; otherwise, $d_{uv} = 0$.

The set of variables are defined as follows:

$$x_{ijk} = \begin{cases} 1 & \text{if arc } (i, j) \text{ is traveled by vehicle } k, \forall i, j \in \{0\} \cup T, k \in P, \\ 0 & \text{otherwise.} \end{cases}$$

$$Z_{ijk} = \begin{cases} 1 & \text{if customer } i \text{ assigned to facility } j \\ & \text{which is visited by vehicle } k, \forall i \in W, j \in T, k \in P, \\ 0 & \text{otherwise.} \end{cases}$$

$$Y_{ijk} = \begin{cases} 1 & \text{if facility } j \\ & \text{is visited by vehicle } k, \forall j \in T, k \in P, \\ 0 & \text{otherwise.} \end{cases}$$

f_{ijk} : the total traveled distance from depot to node $j \in T \cup \{0\}$ when traversing arc (i, j) by vehicle k .

$$\text{Maximize } \sum_{i \in W} \sum_{j \in T} \sum_{k \in P} d_{ij} z_{ijk} \tag{1}$$

s.t.

$$\sum_{i \in T \cup \{0\}} \sum_{j \in T \cup \{0\}} t_{ij} x_{ijk} \leq L_k \quad \forall k \in P, \tag{2}$$

$$\sum_{i \in T \cup \{0\}} x_{ijk} = \sum_{i \in T \cup \{0\}} x_{jik} = y_{jk} \quad \forall j \in T, \forall k \in P, \tag{3}$$

$$\sum_{j \in T \cup \{0\}} x_{j0k} = 1 \quad \forall k \in P, \tag{4}$$

$$\sum_{j \in T \cup \{0\}} x_{0jk} = 1 \quad \forall k \in P, \quad (5)$$

$$z_{ijk} \leq y_{jk} \quad \forall i \in W, \forall j \in T, k \in P, \quad (6)$$

$$\sum_{k \in P} \sum_{j \in T} z_{ijk} \leq 1 \quad \forall i \in W, \quad (7)$$

$$\sum_{k \in P} y_{jk} \leq 1 \quad \forall j \in T, \quad (8)$$

$$\sum_{j \in (T \setminus \{i\}) \cup \{0\}} f_{ijk} - \sum_{j \in (T \setminus \{i\}) \cup \{0\}} f_{jik} = \sum_{j \in T \cup \{0\}} t_{ij} x_{ijk} \quad \forall i \in T, \forall k \in P, \quad (9)$$

$$f_{0ik} = t_{0i} x_{0ik} \quad \forall i \in T, \forall k \in P, \quad (10)$$

$$f_{i0k} \leq L_k x_{i0k} \quad \forall i \in T, \forall k \in P, \quad (11)$$

$$f_{ijk} \leq (L_k - t_{j0}) x_{ijk} \quad \forall (i, j) \in A \& j \neq 0, \forall k \in P, \quad (12)$$

$$f_{ijk} \geq (t_{0i} + t_{ij}) x_{ijk} \quad \forall (i, j) \in A \& i \neq 0, \forall k \in P, \quad (13)$$

$$x_{ijk} + x_{jik} \leq 1 \quad \forall i, j \in T, \forall k \in P, \quad (14)$$

$$x_{ijk} \leq y_{jk} \quad \forall i, j \in T, \forall k \in P, \quad (15)$$

$$f_{ijk} \geq 0 \quad \forall (i, j) \in A, \forall k \in P, \quad (16)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i, j \in T \cup \{0\}, \forall k \in P, \quad (17)$$

$$y_{jk} \in \{0, 1\} \quad \forall j \in T, \forall k \in P, \quad (18)$$

$$z_{ijk} \in \{0, 1\} \quad \forall i \in W, j \in T, \forall k \in P. \quad (19)$$

The objective function (1) maximizes the total number of covered customers. Constraint (2) is to limit the maximum travel time of each vehicle $k \in P$. Constraint (3) implies the connectivity of each route. Constraint sets (4) and (5) assure that all of the routes originate from and terminate at the central depot. Constraint (6) shows that customer i can only be assigned to facility j visited by vehicle k , if $y_{jk} = 1$. By Constraint (7), each customer can be assigned to at most one facility. Constraint (8) indicates that each facility can be visited at most by one vehicle. Constraint sets (9)–(13) prohibit sub-tours which are adopted from (Kara 2011). Constraint (9) is the flow conservation constraint. By using this set of constraints, the total distance traveled from the central depot to node $j \in T \cup \{0\}$ minus the distance traveled from central depot to node $i \in T$, when traveling (i, j) by vehicle k , is equal to the distance from node i to node j . Constraints (10)–(15) are valid inequalities. In particular, Constraint (10) indicates that the total distance from node $i \in T$ to the central depot, when traversing arc $(0, i)$ by vehicle $k \in P$, is equal to the distance between them. Constraint (11) stipulates that the distance from the central depot to the facility $i \in T$ which is traveled by vehicle $k \in P$ cannot exceed L_k . Constraint (12) sets upper bound on f_{ijk} which guarantees the maximum distance traveled to j must not exceed L_k minus the shortest distance that the vehicle k can travel from j to the depot. Constraint (13) sets lower bound on f_{ijk} which shows that the distance traveled to j must be at least equal to the sum of the distance from the depot to i and the distance between i and j . Constraint (14) does not allow sub-tours with only two facilities. Constraint (15) represents that vehicle k

must visit facility j when there is an arc entering to j . Finally, Constraints (16)–(19) specify the type of decision variables.

3 Solution methods

Due to the nature of the problem which makes it NP-hard and because of the applications in emergency situations, it is important to reach qualified solutions in a reasonable computing time. Exact solvers such as CPLEX could not reach optimal solution in a short time, especially for large size instances. Therefore, three heuristic methods, namely ILS, TS and VNS algorithms, are proposed for solving the introduced problem. In addition, we have developed different procedures which are used to generate initial solutions. In the following subsections, the proposed procedures for the initialization phase, the local search procedures which are used in the structure of the three metaheuristic algorithms and the properties of the ILS, TS and VNS are described, respectively.

3.1 Initialization procedure

In this section, we develop two procedures, namely “Sequential” and “Parallel” algorithms, to generate initial solution. In particular, the Sequential algorithm constructs one route at a time, while the Parallel one builds several routes at the same time (Solomon 1987; Potvin and Rousseau 1993; Pisinger and Ropke 2007). In the following, we introduce the characteristics of these algorithms.

3.1.1 Sequential algorithm

In this method, before constructing the routes, facilities are sorted in a descending order of their corresponding scores that will be defined later in this section. Starting from the central depot, the algorithm constructs the first route by visiting the first facility from the ordered set. Upon this step, the corresponding score of each facility is updated and the algorithm selects the next facility to be inserted into its best feasible position of the working route. Essentially, the best position is the place having the minimum extra insertion cost which is imposed to the solution by visiting the new facility. The algorithm keeps adding facilities to the route until it is not possible to visit a new facility because of the route’s length threshold. After completing the first route, the procedure is continued by constructing a new route and the whole procedure stops whenever we have constructed p routes or covered the demand of all customers.

Associated scores for preference of facilities are calculated by using two methods. In the first method, score S_1^j of each facility j is equal to the number of uncovered customers within its coverage distance. In the second method, score S_2^j is computed by applying the following relation.

$$S_2^j = S_1^j / d_j \quad \forall j \in F \quad (20)$$

In this equation, d_j indicates the distance imposed to the solution by inserting facility j into its best position on the tour and F is the set of unrouted facilities. It is worth mentioning that for the first facility, the value of d_j corresponds to its distance from the depot.

3.1.2 Parallel algorithms

In this section, we introduce two types of algorithms, namely “Parallel I” and “Parallel II.” In Parallel I algorithm, facilities are sorted by utilizing one of the methods proposed in Sect. 3.1.1. At the first step, the algorithm selects the p first facilities from ordered set and p routes are constructed at the same time. At each step, the corresponding scores of the remaining facilities are updated and a facility is selected from the ordered set and assigned to the best feasible position generating the minimum extra insertion cost.

In Parallel II algorithm, at the first step, the algorithm selects the p facilities located as far as possible from each other. Following this step, p routes are generated by connecting each facility to the central depot. Routing of the remaining facilities is the same as Parallel I algorithm.

3.2 Local search procedures

In this section, we provide the details of the local search procedures which are used within the structure of the heuristic algorithms. In particular, four procedures have been developed, for which the details are provided in the following.

3.2.1 Swap

Two types of swap move including intra-route and inter-route are used. The goal of the intra-route swap, called “Swap I,” is to minimize the length of the each route k , i.e.,

$\sum_{i \in S_k \cup \{0\}} \sum_{j \in S_k \cup \{0\}} t_{ij} x_{ijk}$. In this definition, S_k is the subset of facilities visited by vehicle $k \in P$. To accelerate the search process, swapping facility $j \in S_k$ with other facilities is considered only for those facilities located within a pre-determined neighborhood radius of j . Neighborhood radius (R) is achieved by applying Eq. (21).

$$R = \theta * \frac{\sum_{i \in T \cup \{0\}} \sum_{j \in T \cup \{0\}} t_{ij}}{((|T| + 1) * (|T| + 2)) / 2} \quad (21)$$

In the above equation, θ is an input parameter taken from $[0, 1]$ for which the corresponding value will be set in Sect. 4.4.

The goal of the second type of swap, called “Swap II,” is to minimize the total length of the routes, i.e., $\sum_{k \in P} \sum_{i \in S_k \cup \{0\}} \sum_{j \in S_k \cup \{0\}} t_{ij} x_{ijk}$, by considering the replacement of each facility $j \in S_k$ with all of the other visited facilities within its neighborhood radius.

3.2.2 Insertion

This move is seeking the maximization of the objective function value, i.e., $\sum_{i \in W} \sum_{j \in T} \sum_{k \in P} d_{ij} z_{ijk}$, by visiting a set of facilities from F which can cover at least one uncovered customer. For these facilities, the best insertion position is specified and the facility with the largest value of S_2^j (see, Eq. 20) is added to its best feasible location. By inserting the first facility, the set F and the value of the objective function are updated. This procedure continues until the possibility of inserting all members of F is examined.

3.2.3 Deletion

Since some facilities cover common customers, by deleting a routed facility, all of its covered customers may be served by other routed facilities. So, by removing a routed facility, without any changes in the objective function value, i.e., $\sum_{i \in W} \sum_{j \in T} \sum_{k \in P} d_{ij} z_{ijk}$, the length of the routes may be decreased. Therefore, the possibility to insert other facilities may increase and the objective function will increase as well. This move is done for all the routed facilities, and the facility with the most reduction in the total length of the routes is selected.

3.2.4 Extraction–Insertion

This procedure includes two types. In the first one “Extraction–Insertion I,” the goal is to decrease the tour’s length by repositioning the routed facilities. In particular, a facility is extracted from its location and reinserted to the place on the same route with the minimum reinsertion cost. Applying this procedure is tested for all of the routed facilities, and a facility with the most reduction cost is selected.

In the second type of this procedure “Extraction–Insertion II,” the goal is to increase the value of the objective function, or in the case that the objective value could not be improved anymore, it is to decrease the tours’ length without reduction in the objective value. In particular, it is done by removing routed facilities from their locations and adding unrouted facilities into the best position on the available routes. Starting from the first constructed route, the first visited facility is extracted from its location and a set of unrouted facilities which covers at least one of the uncovered customers is inserted into the solution. At each step, the added facility is the one having the minimum insertion cost. If this substitution leads to an improvement in the objective function or it can decrease the total length without any reduction in the objective function, the corresponding move is saved. Applying this procedure is checked for the rest of the routed facilities. Finally, the substitution with the best improvement is selected for executing the procedure.

3.3 Iterated local search

In this section, we describe the ILS procedure which is used to solve the proposed problem (Algorithm 1). The ILS is a metaheuristic algorithm that combines local search

and perturbation procedures to search the feasible region more efficiently (Lourenço et al. 2003). In particular, this algorithm has been successfully applied to solve similar problems such as TOP (Vansteenwegen et al. 2009; Gunawan et al. 2015; Zachariadis and Kiranoudis 2011). For this reason, we use the ILS as one of the proposed methods for solving our problem. In this algorithm, a solution is created by following one of the proposed procedures in Sect. 3.1. The algorithm is composed of two loops. The inner loop is to improve the solution's cost by applying local search procedures. Essentially, at each iteration, a local search procedure is selected by using the roulette wheel technique (see Sect. 3.3.1). Before entering the outer loop, as long as the solution can be improved, the LS algorithm is applied. The pseudocode of the LS procedure is depicted in Algorithm 2. In fact, we tested several orders of the moves and the current order is the best combination. In this combination, at first we utilize the swap procedures followed by the extraction–insertion procedures. The reason of using the deletion and insertion procedures is that they can further improve the solutions obtained by applying the two mentioned procedures (i.e., swap and extraction–insertion), and this is why we call them after each execution of the swap and extraction–insertion procedures. The whole procedure is iterated for “*iter1*” iterations. In addition, during the execution of the outer loop, the goal is to escape from local optima by utilizing the perturbation procedure. The outer loop is repeated “*iter0*” iterations. The corresponding values for *iter0* and *iter1* will be set in Sect. 4.4.

Algorithm 1 The pseudocode of the ILS algorithm

```

1: Input InitialSolution;
2: CurrentSolution = InitialSolution;
3: CurrentSolution = LS (CurrentSolution);
4: BestSolution = CurrentSolution;
5:  $i_0 = 0$ ;
6: while  $i_0 \leq \textit{iter0}$  do
7:    $i_0 = i_0 + 1$ ;
8:    $i_1 = 0$ ;
9:   While  $i_1 \leq \textit{iter1}$  do
10:     $i_1 = i_1 + 1$ ;
11:    move = select one of the local search procedures by applying the roulette wheel technique;
12:    CurrentSolution = move(CurrentSolution);
13:    if selected move improves the CurrentSolution then
14:      update CurrentSolution;
15:      if (cost(CurrentSolution) < cost (BestSolution)) then
16:        CurrentSolution = LS (CurrentSolution);
17:        BestSolution = CurrentSolution;
18:      End if
19:    else
20:      CurrentSolution = select the best feasible neighborhood leading to the least cost increment;
21:    while True do
22:      CurrentSolution = Deletion(CurrentSolution);
23:      CurrentSolution = Insertion(CurrentSolution);
24:      if (cost(CurrentSolution) >= cost(BestSolution)) then
25:        break;
26:      else
27:        BestSolution = CurrentSolution;
28:      End while
29:    End while
30:    CurrentSolution = BestSolution;
31:    CurrentSolution = Perturbation(CurrentSolution);
32: End while
33: Output BestSolution;

```

3.3.1 Roulette wheel

In each iteration of the ILS algorithm, one of the local search procedures is selected based on roulette wheel (see line 11 in Algorithm 1). In this method, in the first iteration the possibility of choosing a move is equal to $\frac{1}{n}$ where n is the total number of moves. The possibility of selecting different procedures in later iterations is proportional to the effectiveness of the procedures in the previous iterations. Essentially, if the best solution is improved by applying a specific procedure, the possibility of selecting the corresponding procedure is increased in the subsequent iteration.

3.3.2 Perturbation phase

To escape from local optima, after every “*iterI*” iterations of the inner loop, the perturbation procedure is applied on the best found solution (see line 31 in Algorithm 1). In this procedure at first, ρ percent of the visited facilities are selected, randomly, and removed from their corresponding routes and added to the set F (set of unrouted facilities). Each time a facility is removed from the route, the assignment of customers is updated. In the second step, a facility is selected from F , randomly, and is added into its best feasible insertion position. The preliminary results show in case that we use the same set of customers in the structure of the perturbed solution, there will be a high probability to get stuck in a local optima solution by applying the local search procedure. In fact, by inserting the facilities randomly, we try to escape from local optima and expand the search space. After inserting the facility, the objective function and the subset F are updated. This step is iterated as long as we are not able to insert a new facility.

Algorithm 2 The pseudocode of the LS algorithm

```

1: Input CurrentSolution;
2: while True do
3:   CurrentSolution = Swap I(CurrentSolution);
4:   CurrentSolution = Deletion(CurrentSolution);
5:   CurrentSolution = Insertion(CurrentSolution);
6:   CurrentSolution = Swap II(CurrentSolution);
7:   CurrentSolution = Deletion(CurrentSolution);
8:   CurrentSolution = Insertion(CurrentSolution);
9:   CurrentSolution = ExtractionInsertion I(CurrentSolution);
10:  CurrentSolution = Deletion(CurrentSolution);
11:  CurrentSolution = Insertion(CurrentSolution);
12:  CurrentSolution = ExtractionInsertion II(CurrentSolution);
13:  CurrentSolution = Deletion(CurrentSolution);
14:  CurrentSolution = Insertion(CurrentSolution);
15:  if (cost (CurrentSolution) <= (cost (BestSolution))) then
16:    BestSolution = CurrentSolution;
17:  else
18:    break;
19:
20: End
21: Output BestSolution;

```

3.4 TS algorithm

TS is a metaheuristic algorithm introduced by Glover (1989). To escape from local optimum solution, this algorithm always moves to the best neighbor solution, even if the corresponding neighbor solution does not lead to an improvement in the cost of the current solution.

The proposed TS algorithm starts from the best solution obtained by utilizing the LS procedure (see Algorithm 2) and in each iteration selects a local search procedure by using the roulette wheel technique to explore the current solution's neighborhoods. For each selected neighborhood, the "best neighbor" with the most improvement in the cost is selected. If the implemented move is not tabu, it is selected as the new incumbent solution. Otherwise, the move is applied only if it has improved the best solution that has been found. In addition, we execute the diversification phase after "*iter4*" iterations without any improvements in the cost of the best solution. Finally, the intensification phase is applied after each improvement of the best solution which is discussed in Sect. 3.4.1. The pseudocode of the TS algorithm is provided in Algorithm 3 for which the details are given in the following subsections.

3.4.1 Short-term memory

Upon applying each procedure, the attributes of the obtained solution must be considered as tabu in order to prevent choosing them for a specified number of iterations. In this paper, we have two types of tabu list as the short memory. In the first tabu list, the attributes of the swap I and the swap II are stored, while the second list contains the attributes of the Extraction–Insertion I and the Extraction–Insertion II. Attributes of the first tabu list are stored as (i, j) which indicates swapping facilities i and j are forbidden for a fixed number of iterations. Attributes of the second list are the indices of the facilities extracted from or inserted to one of the routes through the corresponding extraction–insertion procedure. So, if the facility has inserted (extracted), it will not be extracted (inserted) for the following τ iterations.

Algorithm 3 The pseudocode of the TS algorithm

```

1: Input InitialSolution;
2: CurrentSolution = InitialSolution;
3: CurrentSolution = LS (CurrentSolution);
4: BestSolution = CurrentSolution;
5:  $i_2 = 0$ ;
6: while  $i_2 \leq \text{iter2}$  do
7:    $i_2 = i_2 + 1$ ;
8:    $i_3 = 0$ ;
9:    $i_4 = 0$ ;
10:  While  $i_3 \leq \text{iter3}$  do
11:     $i_3 = i_3 + 1$ ;
12:    TempBestSolution = BestSolution;
13:    move = select one of the local search procedures by applying the roulette wheel technique;
14:    if selected move improves CurrentSolution then
15:      select the best neighbor;
16:      if neighborhood isn't tabu then
17:        update tabu list;
18:      else if aspiration criterion is satisfied then
19:        BestSolution = CurrentSolution;
20:        update  $r_{ij}$ ;
21:        update  $\sigma$ ;
22:      else
23:        break;
24:      End if
25:      if  $\text{cost}(\text{CurrentSolution}) < \text{cost}(\text{BestSolution})$  then
26:        BestSolution = LS (CurrentSolution);
27:        update  $r_{ij}$ ;
28:        update  $\sigma$ ;
29:      End if
30:    else
31:      CurrentSolution = select the best feasible neighborhood leading to the least cost increment;
32:      update tabu list;
33:      update CurrentSolution;
34:      while True do
35:        CurrentSolution = Deletion(CurrentSolution);
36:        CurrentSolution = Insertion(CurrentSolution);
37:        if  $(\text{cost}(\text{CurrentSolution}) \geq \text{cost}(\text{BestSolution}))$  then
38:          break;
39:        else
40:          BestSolution = CurrentSolution;
41:        End while
42:      End if
43:      if BestSolution == TempBestSolution then
44:         $i_4 = i_4 + 1$ ;
45:        if  $i_4 \geq \text{iter4}$  do
46:           $i_5 = i_3 + \text{iter5}$ ;
47:          CurrentSolution = BestSolution;
48:          while  $i_3 \leq i_5$  do
49:             $i_3 = i_3 + 1$ ;
50:            CurrentSolution = Diversification (CurrentSolution);
51:          End while
52:        End if
53:      End if
54:    End if
55:  End while
56:  CurrentSolution = BestSolution;
57:  CurrentSolution = Perturbation(CurrentSolution);
58: End while
59: Output BestSolution;

```

3.4.2 Long-term memory

Long-term memory is based on the frequency of edges which appear in different solutions. During search process, whenever we have an improvement in the cost of the best solution, the corresponding edges of the improved solution are kept in long-term memory to be used in the diversification phase. In particular, diversification is

implemented after “*iter4*” iterations without any improvements. In this phase, the objective function of solution s , i.e., $f(s)$, is decreased by imposing a penalty which is proportional to the frequency of edges in s . To do so, a new objective function is calculated based on Eq. (22). In this equation, $f(s)$ is the number of covered customers by visited facilities, i.e., $\sum_{i \in W} \sum_{j \in T} \sum_{k \in P} d_{ij} z_{ijk}$.

$$g(s) = f(s) * (1 - r_{ij} / \sigma) \quad (22)$$

In addition, r_{ij} is the number of improved solutions that contain edge (i, j) and σ denotes the total number of improved solutions. In the diversification phase, selection of the local search procedures is based on the roulette wheel technique and evaluation criterion for selecting neighborhoods is $g(s)$ instead of $f(s)$. In particular, by using $g(s)$, solutions consisting of edges with less frequencies during the search history will have a lower penalty. Diversification is iterated for “*iter5*” iterations, and in each iteration, the neighborhood which has the best value of $g(s)$ is selected. Finally, in each iteration of the TS, if the selected move improves the best solution, local search phase is applied as the intensification strategy.

3.5 VNS algorithm

Proposed by Mladenović and Hansen (1997), the VNS is a metaheuristic algorithm that explicitly applies a strategy based on dynamically changing the neighborhood structures. The general framework of the VNS algorithm for the introduced problem is given in Algorithm 4. The algorithm starts to construct an initial solution (*CurrentSolution*) by applying the initialization procedure (see Sect. 3.1). It then improves upon this initial solution by applying the LS procedure. The improvement of the solution continues in a loop until the termination criterion is met. The loop contains the perturbation procedure and the LS procedures. If a solution is improved by applying the perturbation and local search procedures it will be accepted as the incumbent solution. At each iteration of the VNS algorithm, we apply the perturbation procedure in order to dynamically expand the neighborhood structure. In particular, the algorithm generates a solution which is in the neighborhood size ρ of the *CurrentSolution*, i.e., $\text{Perturbation}(\text{CurrentSolution}, \rho)$. In this definition, ρ is the percentage of customers which is extracted by utilizing the perturbation procedure (see Sect. 3.3.2).

4 Computational results

The proposed algorithms have been coded in Visual C++ 2010, and the mathematical model has been solved by using ILOG CPLEX 12.3. All tests have been performed on a PC Intel Core i7 2.93 GHz processor and 3.49 GB of RAM. This section contains detailed computational results of running CPLEX, TS, ILS and VNS on the modified samples.

Algorithm 4 The pseudocode of the VNS algorithm

```

1: Input InitialSolution;
2: CurrentSolution = InitialSolution;
3: CurrentSolution = LS (CurrentSolution);
4: BestSolution = CurrentSolution;
5:  $i_6 = 0$ ;
6:  $\rho = \rho_1$ ;
7: while  $i_6 \leq \text{iter6}$  do
8:    $i_6 = i_6 + 1$ ;
9:    $i_7 = 0$ ;
10:  TempBestSolution = BestSolution;
11:  While  $i_7 \leq \text{iter7}$  do
12:    move = select one of the local search procedures by applying the roulette wheel technique;
13:    CurrentSolution = move(CurrentSolution);
14:    if selected move improves the CurrentSolution then
15:      update CurrentSolution;
16:      if (cost(CurrentSolution) < cost (BestSolution)) then
17:        CurrentSolution = LS (CurrentSolution);
18:        BestSolution = CurrentSolution;
19:      End if
20:    else
21:      CurrentSolution = select the best feasible neighborhood leading to the least cost increment;
22:       $i_7 = i_7 + 1$ ;
23:    while True do
24:      CurrentSolution = Deletion(CurrentSolution);
25:      CurrentSolution = Insertion(CurrentSolution);
26:      if (cost(CurrentSolution) >= cost(BestSolution)) then
27:        break;
28:      else
29:        BestSolution = CurrentSolution;
30:      End while
31:    End while
32:    CurrentSolution = BestSolution;
33:    if (cost(TempBestSolution) > (cost (BestSolution))) then
34:       $\rho = \rho_1$ ;
35:    else
36:      if ( $\rho < \rho_{max}$ )
37:         $\rho = \rho + \gamma$ ;
38:      End if
39:    CurrentSolution = Perturbation(CurrentSolution,  $\rho$ );
40:  End while
41: Output BestSolution;

```

4.1 Data generation

Several instances have been generated to evaluate the effectiveness of the proposed algorithms. These samples are randomly generated and classified into three sets including small, medium and large size data. In particular, we use the instances from the TSP library having 52, 76, 100, 150, 200, 318, 417, 575, 657 and 724 nodes (Reinelt 1991). In generated instances each facility can cover up to 5 nearest customers, randomly. By using each instance, we have generated 3 new instances in which for each sample, 50, 60 and 70% of the nodes are considered as customers ($|W|$). Three different scenarios, namely 2, 3 and 4, are considered for the number of vehicles ($|P|$). Finally, three values are proposed for L_k as follows:

$$L_k = \frac{\alpha}{|P|} \times |T| \times \frac{\sum_{i \in T \cup \{0\}} \sum_{j \in T \cup \{0\}} t_{ij}}{((|T|) * (|T| + 1)) / 2} \quad (23)$$

In this equation $\alpha \in \{1, 0.9, 0.8\}$. As a result, totally we have $10 \times 3 \times 3 \times 3 = 270$ instances. The instances are divided into the three groups. In particular, the groups with small and medium instances contain instances having up to 100 and 200 nodes, respectively, while the other instances belong to the group with large instances.

4.2 CPLEX results

ILOG CPLEX has been used to run the mathematical model in Sect. 2. In particular, for each instance 5 h of computing time is put on the maximum run-time of CPLEX and the results are reported in Tables 1, 2 and 3. In these tables, number of nodes, number of facilities, number of customers, number of vehicles and the maximum travel length are provided in columns 2–6, respectively. The corresponding objective function is provided in the column “*Obj.*” and the corresponding run-time of CPLEX (in seconds) is given in the column labeled by “Time”. Finally, the optimality gap (in percentage) is provided in the column “Gap”. Totally, 78 small instances (out of 81), 34 medium instances (out of 54) and 32 large instances (out of 135) have been solved to optimality within the given time limit. For those instances that CPLEX is not able to reach the optimal solution, the average optimality gap is 2.28% for the small instances, while this is 3.84% for the medium instances. In addition, the average computing time for the small and medium size instances is 905.75 and 7803.84 s, respectively. As it is shown in Table 3, CPLEX is able to reach the feasible solution only in 38 instances (out of 135) of the large size data and it fails to reach a feasible solution for the other 97 instances. The average computing time for 38 instances for which CPLEX can achieve a feasible solution is 1079.38 s. There are some instances for which the optimality gap is a positive value, while CPLEX has not reached the time limit. In these instances, the computer runs out of memory before finding the optimal solution and the best found solution is reported in the corresponding position in Table 3.

According to the applications of this problem in critical situations and emergencies and due to the fact that in real-world situations we may face the larger size problems, we should use heuristic methods to accelerate obtaining high-quality solutions in a reasonable computing time.

4.3 Analyzing the performance of the initialization algorithms

As discussed in Sect. 3.1, we have proposed three methods, namely Sequential, Parallel I and Parallel II, for generating initial solutions. In addition, we developed two scoring methods, i.e., S_1^j and S_2^j , for preference of facilities. Therefore, we have 6 different scenarios to generate initial solutions.

Since CPLEX fails to reach a feasible solution for several large instances, we have only used the results obtained by CPLEX for the small and medium size data in order to analyze the performance of the initialization procedures. Results by utilizing each of the 6 scenarios on small and medium instances are given in Table 4. In particular, for each scenario the overall average gap for available instances, with respect to the results obtained by CPLEX, is given in the “Avg. gap” row. Based on the results, the

Table 1 CPLEX results for small instances of the TCMCRP

Instance	VI	VT	WI	PI	L_k	Obj.	Time	Gap	Instance	VI	VT	WI	PI	L_k	Obj.	Time	Gap
1	52	15	36	2	1368.03	15	0.03	0.00	42	76	30	45	3	9367.66	12	0.09	0.00
2	52	15	36	2	1231.26	12	0.05	0.00	43	76	30	45	4	8782.18	12	0.08	0.00
3	52	15	36	2	1090.05	15	0.03	0.00	44	76	30	45	4	7903.96	12	0.08	0.00
4	52	15	36	3	912.05	11	0.06	0.00	45	76	30	45	4	7025.74	7	0.05	0.00
5	52	15	36	3	820.84	9	0.05	0.00	46	76	37	38	2	22,400.71	17	4.61	0.00
6	52	15	36	3	729.64	8	0.03	0.00	47	76	37	38	2	20,160.64	17	1.80	0.00
7	52	15	36	4	684.04	8	0.05	0.00	48	76	37	38	2	17,920.57	15	2.77	0.00
8	52	15	36	4	615.63	8	0.05	0.00	49	76	37	38	3	14,933.80	14	1.06	0.00
9	52	20	31	4	547.23	1	0	0.00	50	76	37	38	3	13,440.42	14	0.25	0.00
10	52	20	31	2	1707.37	15	3.50	0.00	51	76	37	38	3	11,974.04	12	0.16	0.00
11	52	20	31	2	1536.63	15	0.92	0.00	52	76	37	38	4	11,200.35	12	0.19	0.00
12	52	20	31	2	1365.89	14	1.30	0.00	53	76	37	38	4	10,080.32	11	0.14	0.00
13	52	20	31	3	1138.24	13	0.09	0.00	54	76	37	38	4	8960.26	9	0.09	0.00
14	52	20	31	3	1024.42	13	0.06	0.00	55	100	29	70	2	6193.82	36	350.05	0.00
15	52	20	31	3	910.59	12	0.55	0.00	56	100	29	70	2	5574.44	36	5.56	0.00
16	52	20	31	4	853.68	10	0.08	0.00	57	100	29	70	2	4955.06	36	3.26	0.00
17	52	20	31	4	768.31	9	0.08	0.00	58	100	29	70	3	4129.21	36	0.45	0.00
18	52	20	31	4	682.95	9	0.09	0.00	59	100	29	70	3	3716.29	36	0.25	0.00

Table 1 continued

Instance	VI	IT	WI	IPI	L_k	Obj.	Time	Gap	Instance	VI	IT	WI	IPI	L_k	Obj.	Time	Gap
19	52	25	26	2	1988.24	20	29.67	0.00	60	100	29	70	3	3303.37	21	0.38	0.00
20	52	25	26	2	1789.42	19	20.30	0.00	61	100	29	70	4	3096.91	21	0.20	0.00
21	52	25	26	2	1590.59	18	3.48	0.00	62	100	29	70	4	2787.22	16	0.13	0.00
22	52	25	26	3	1325.49	17	56.84	0.00	63	100	29	70	4	2477.53	14	0.08	0.00
23	52	25	26	3	1192.94	15	1.84	0.00	64	100	39	60	2	8326.06	51	91.84	0.00
24	52	25	26	3	1060.39	15	0.17	0.00	65	100	39	60	2	7493.47	51	2.3	0.00
25	52	25	26	4	994.12	14	0.61	0.00	66	100	39	60	2	6660.87	49	10,564.23	0.00
26	52	25	26	4	894.71	11	21.95	0.00	67	100	39	60	3	5550.72	48	18,000.00	2.08
27	52	25	26	4	795.30	12	0.13	0.00	68	100	39	60	3	4995.65	40	5433.92	0.00
28	76	22	53	2	11,960.43	12	0.06	0.00	69	100	39	60	3	4440.58	38	314.39	0.00
29	76	22	53	2	10,764.43	12	0.06	0.00	70	100	39	60	4	4163.04	34	9.91	0.00
30	76	22	53	2	9568.38	10	0.05	0.00	71	100	39	60	4	3746.04	32	0.84	0.00
31	76	22	53	3	7973.65	10	0.06	0.00	72	100	39	60	4	3330.43	27	2.16	0.00
32	76	22	53	3	7176.29	7	0.05	0.00	73	100	49	50	2	10,458.99	45	7.53	0.00
33	76	22	53	3	6378.92	7	0.03	0.00	74	100	49	50	2	9413.09	45	26.97	0.00
34	76	22	53	4	5980.24	5	0.03	0.00	75	100	49	50	2	8367.19	45	36.72	0.00
35	76	22	53	4	5382.22	5	0.03	0.00	76	100	49	50	3	6972.66	45	1018.36	0.00
36	76	22	53	4	4774.19	5	0.03	0.00	77	100	49	50	3	6275.93	45	1174.98	0.00
37	76	30	45	2	17,564.36	16	2.64	0.00	78	100	49	50	3	5578.13	42	18,000.00	2.38
38	76	30	45	2	15,807.92	16	1.08	0.00	79	100	49	50	4	5229.49	42	18,000.00	2.38
39	76	30	45	2	14,051.49	13	14.11	0.00	80	100	49	50	4	4706.55	37	126.72	0.00
40	76	30	45	3	11,709.57	13	0.22	0.00	81	100	49	50	4	4183.60	33	11.57	0.00
41	76	30	45	3	10,583.61	13	0.13	0.00									

Table 2 CPLEX results for medium instances of the TCMCRP

Instance	VI	VTI	WI	WPI	L _k	Obj.	Time	Gap	Instance	VI	VTI	WI	WPI	L _k	Obj.	Time	Gap
82	150	44	105	2	9210.94	72	454.5	0.00	109	200	59	140	2	12,513.29	104	89.2	0.00
83	150	44	105	2	8289.84	72	1729.3	0.00	110	200	59	140	2	11,261.96	104	173.22	0.00
84	150	44	105	2	7368.75	70	18,000.00	2.86	111	200	59	140	2	10,010.63	102	18,000.00	1.96
85	150	44	105	3	6140.62	69	18,000.00	4.35	112	200	59	140	3	8342.19	92	18,000.00	13.04
86	150	44	105	3	5526.56	69	55.7	0.00	113	200	59	140	3	7507.97	104	10,308.03	0.00
87	150	44	105	3	4912.50	64	18,000.00	1.56	114	200	59	140	3	6673.75	101	18,000.00	1.98
88	150	44	105	4	4605.47	66	18,000.00	1.52	115	200	59	140	4	6256.64	100	18,000.00	2.74
89	150	44	105	4	4144.92	58	18,000.00	2.58	116	200	59	140	4	5630.98	88	18,000.00	2.27
90	150	44	105	4	3684.37	49	1555.25	0.00	117	200	59	140	4	5005.31	88	18,000.00	2.27
91	150	59	90	2	12,364.81	71	45.13	0.00	118	200	79	120	2	16,673.37	107	60.19	0.00
92	150	59	90	2	11,128.36	71	613.53	0.00	119	200	79	120	2	15,006.03	107	89.13	0.00
93	150	59	90	2	9891.85	71	2193.17	0.00	120	200	79	120	2	13,338.69	107	261.84	0.00
94	150	59	90	3	8243.20	70	18,000.00	1.43	121	200	79	120	3	11,115.58	107	9167.97	0.00
95	150	59	90	3	7418.88	71	6267.77	0.00	122	200	79	120	3	10,004.02	103	18,000.00	3.88

Table 2 continued

Instance	IVI	ITI	IWI	IPI	L_k	Obj.	Time	Gap	Instance	IVI	ITI	IWI	IPI	L_k	Obj.	Time	Gap
96	150	59	90	3	6594.56	69	13,768.27	2.90	123	200	79	120	3	8892.46	100	18,000.00	7.00
97	150	59	90	4	6182.40	71	2920.02	0.00	124	200	79	120	4	8336.68	104	18,000.00	2.89
98	150	59	90	4	5564.18	71	5403.89	0.00	125	200	79	120	4	7503.01	103	18,000.00	3.88
99	150	59	90	4	4945.92	65	18,000.00	4.43	126	200	79	120	4	6669.35	95	18,000.00	11.58
100	150	74	75	2	15,315.62	64	32.13	0.00	127	200	99	100	2	20,883.45	94	101.12	0.00
101	150	74	75	2	13,784.06	64	23.88	0.00	128	200	99	100	2	18,795.10	94	177.72	0.00
102	150	74	75	2	12,252.49	64	174.27	0.00	129	200	99	100	2	16,706.76	94	254.14	0.00
103	150	74	75	3	10,210.41	64	176.11	0.00	130	200	99	100	3	13,922.33	94	96.47	0.00
104	150	74	75	3	9189.37	64	278.77	0.00	131	200	99	100	3	12,530.07	94	459.86	0.00
105	150	74	75	3	8186.33	63	18,000.00	1.59	132	200	99	100	3	11,137.84	94	646.31	0.00
106	150	74	75	4	7657.81	64	3799.25	0.00	133	200	99	100	4	10,441.72	94	1057.53	0.00
107	150	74	75	4	6892.03	64	417,913	0.00	134	200	99	100	4	9397.55	94	8875.07	0.00
108	150	74	75	4	6126.25	64	7619.86	0.00	135	200	99	100	4	8553.38	94	3986.34	0.00

Table 3 CPLEX results for large instances of the TCMCRP

Instance	VI	IT	IWI	IPI	L_k	Obj.	Time	Gap	Instance	VI	IT	IWI	IPI	L_k	Obj.	Time	Gap
1	318	158	159	2	5857.933	40	146.17	0.00	42	417	166	250	2	7243.531	-	-	-
2	318	158	159	3	22,073.87	37	1626.64	0.00	43	417	166	250	3	16,096.74	-	-	-
3	318	158	159	4	14,715.91	-	-	-	44	417	166	250	4	10,731.16	-	-	-
4	318	158	159	2	11,036.93	41	531.50	0.00	45	417	166	250	2	8048.368	-	-	-
5	318	158	159	3	24,833.1	40	795.27	0.05	46	417	125	291	3	9372.693	75	87.82	0.00
6	318	158	159	4	16,555.4	-	-	-	47	417	125	291	4	6248.462	72	387.45	0.00
7	318	158	159	2	12,416.55	-	-	-	48	417	125	291	2	4686.347	58	2604.34	0.29
8	318	158	159	3	27,592.33	-	-	-	49	417	125	291	3	10,544.28	73	88.23	0.00
9	318	158	159	4	18,394.89	-	-	-	50	417	125	291	4	7029.52	73	216.63	0.00
10	318	127	190	2	13,796.17	36	145.19	0.00	51	417	125	291	2	5272.14	-	-	-
11	318	127	190	3	17,407.31	41	245.66	0.00	52	417	125	291	3	11,715.87	75	76.69	0.00
12	318	127	190	4	11,604.87	33	698.38	0.12	53	417	125	291	4	7810.577	-	-	-
13	318	127	190	2	8703.653	39	95.9	0.00	54	417	125	291	2	7243.531	-	-	-
14	318	127	190	3	19,583.22	43	190.83	0.00	55	575	287	287	3	3624.824	-	-	-
15	318	127	190	4	13,055.48	-	-	-	56	575	287	287	4	2416.549	-	-	-
16	318	127	190	2	9791.609	40	62.1	0.00	57	575	287	287	2	1812.412	-	-	-
17	318	127	190	3	21,759.13	-	-	-	58	575	287	287	3	4077.927	-	-	-
18	318	127	190	4	14,506.09	-	-	-	59	575	287	287	4	2718.618	-	-	-
19	318	95	222	2	10,879.57	43	44.31	0.00	60	575	287	287	2	2038.963	-	-	-
20	318	95	222	3	10,719.94	38	165.72	0.00	61	575	287	287	3	4531.03	-	-	-
21	318	95	222	4	7146.625	43	546.76	0.00	62	575	287	287	4	3020.687	-	-	-
22	318	95	222	2	5359.969	41	67.45	0.00	63	575	287	287	2	2265.515	-	-	-
23	318	95	222	3	12,059.93	37	126.49	0.00	64	575	229	345	3	2589.767	27	1360.94	0.00
24	318	95	222	4	8039.954	37	304.4	0.054	65	575	229	345	4	1726.511	-	-	-

Table 3 continued

Instance	VI	IT1	IWI	IPI	L_k	Obj.	Time	Gap	Instance	VI	IT1	IWI	IPI	L_k	Obj.	Time	Gap
25	318	95	222	2	6029,965	41	70.14	0.00	66	575	229	345	2	1294,883	-	-	-
26	318	95	222	3	13,399,92	40	154.47	0.00	67	575	229	345	3	2913,487	-	-	-
27	318	95	222	4	8933,282	-	-	-	68	575	229	345	4	1942,325	-	-	-
28	417	208	208	2	17,827,96	58	255.73	0.00	69	575	229	345	2	1456,744	-	-	-
29	417	208	208	3	11,885,31	54	598.31	0.00	70	575	229	345	3	3237,208	-	-	-
30	417	208	208	4	8913,982	-	-	-	71	575	229	345	4	2158,139	-	-	-
31	417	208	208	2	20,056,46	58	244.86	0.00	72	575	229	345	2	1618,604	-	-	-
32	417	208	208	3	13,370,97	-	-	-	73	575	172	402	3	1732,686	27	239.37	0.00
33	417	208	208	4	10,028,23	-	-	-	74	575	172	402	4	1155,124	26	493.07	0.00
34	417	208	208	2	22,284,96	-	-	-	75	575	172	402	2	866,3432	-	-	-
35	417	208	208	3	14,856,64	-	-	-	76	575	172	402	3	1949,272	30	201.3	0.00
36	417	208	208	4	11,142,48	-	-	-	77	575	172	402	4	1299,515	-	-	-
37	417	166	250	2	12,877,39	62	18,000	0.0161	78	575	172	402	2	974,636	-	-	-
38	417	166	250	3	8584,926	64	3319.31	0.00	79	575	172	402	3	2165,858	19	335.1	0.37
39	417	166	250	4	6438,694	-	-	-	80	575	172	402	4	1443,905	-	-	-
40	417	166	250	2	14,487,06	63	211.125	0.00	81	575	172	402	2	1082,929	-	-	-
41	417	166	250	3	9658,041	-	-	-	-	-	-	-	-	-	-	-	-
82	657	328	328	2	30,838,17	-	-	-	109	724	361	362	2	40,848,98	-	-	-
83	657	328	328	3	20,558,78	-	-	-	110	724	361	362	3	27,232,65	-	-	-
84	657	328	328	4	15,419,08	-	-	-	111	724	361	362	4	20,424,49	-	-	-
85	657	328	328	2	34,692,94	-	-	-	112	724	361	362	2	45,955,09	-	-	-
86	657	328	328	3	23,128,62	-	-	-	113	724	361	362	3	30,636,73	-	-	-
87	657	328	328	4	17,346,47	-	-	-	114	724	361	362	4	22,977,55	-	-	-

Table 3 continued

Instance	VI	IT1	IW1	IPI	L_k	Obj.	Time	Gap	Instance	VI	IT1	IW1	IPI	L_k	Obj.	Time	Gap
88	657	328	328	2	38,547.71	-	-	-	115	724	361	362	2	1061.219	-	-	-
89	657	328	328	3	25,698.47	-	-	-	116	724	361	362	3	34,040.81	-	-	-
90	657	328	328	4	19,273.85	-	-	-	117	724	361	362	4	25,530.61	-	-	-
91	657	262	394	2	22,208.68	76	3358.61	0.00	118	724	289	434	2	29,871.47	-	-	-
92	657	262	394	3	14,805.79	-	-	-	119	724	289	434	3	19,914.32	-	-	-
93	657	262	394	4	11,104.34	-	-	-	120	724	289	434	4	14,935.74	-	-	-
94	657	262	394	2	24,984.76	-	-	-	121	724	289	434	2	33,605.41	-	-	-
95	657	262	394	3	16,656.51	-	-	-	122	724	289	434	3	22,403.6	-	-	-
96	657	262	394	4	12,492.38	-	-	-	123	724	289	434	4	16,802.7	-	-	-
97	657	262	394	2	27,760.85	-	-	-	124	724	289	434	2	37,339.34	-	-	-
98	657	262	394	3	18,507.23	-	-	-	125	724	289	434	3	24,892.89	-	-	-
99	657	262	394	4	13,880.42	-	-	-	126	724	289	434	4	18,669.67	-	-	-
100	657	197	459	2	14,375.04	63	1693.45	0.00	127	724	217	506	2	19,912.23	-	-	-
101	657	197	459	3	9583.36	-	-	-	128	724	217	506	3	13,274.82	-	-	-
102	657	197	459	4	7187.52	-	-	-	129	724	217	506	4	9956.117	-	-	-
103	657	197	459	2	16,171.92	65	1226.75	0.00	130	724	217	506	2	22,401.26	-	-	-
104	657	197	459	3	10,781.28	-	-	-	131	724	217	506	3	14,934.17	-	-	-
105	657	197	459	4	8085.959	-	-	-	132	724	217	506	4	11,200.63	-	-	-
106	657	197	459	2	17,968.8	-	-	-	133	724	217	506	2	24,890.29	-	-	-
107	657	197	459	3	11,979.2	-	-	-	134	724	217	506	3	16,593.53	-	-	-
108	657	197	459	4	8984.399	-	-	-	135	724	217	506	4	12,445.15	-	-	-

Table 4 Comparing the performance of different initialization procedures

Initial solution type	Sequential (S_1^j)	Sequential (S_2^j)	Parallel I (S_1^j)	Parallel I (S_2^j)	Parallel II (S_1^j)	Parallel II (S_2^j)
Avg. gap	2.32	2.16	1.73	2.77	6.82	14.2
Avg. time	0.01	0.01	0.01	0.01	0.01	0.01

Parallel I algorithm with S_1^j scoring method has the minimum average gap, while the worst gap is obtained by Parallel II algorithm with S_2^j scoring method. Finally, the row labeled by “Avg. time” shows, on average, different methods need almost the same computing time to achieve an initial solution.

4.4 Parameter tuning

The proposed heuristic algorithms contain a set of parameters to be tuned. For each algorithm, Table 5 gives the assigned values for the involved parameters and the final selected values. To set the values, we have chosen one-third of the small and medium size problems as the set of benchmark instances to run the experiments. In particular, all the combinations of the parameters are tested and the best combination leading to the best average performance is selected for the corresponding algorithm. To analyze the results, we have used Wilcoxon nonparametric test with 95% of confidence. This test is presented in Wilcoxon (1945) and used instead of t test when the probability distribution of samples is unknown. For each algorithm, to select the best combination of parameters we select the combinations having the best and worst overall performance for the set of benchmark instances. Wilcoxon test shows that the difference between the results of these two combinations is not statistically significant. As a result, for each algorithm the combination having the minimum computing time is chosen to run the final experiments.

4.5 ILS results for small and medium instances

Each of the initialization procedures is used to run the ILS algorithm. Essentially, for each of the initial solutions we run the ILS algorithm five times with five different random seeds and the average and the best performance of the ILS algorithm are provided in Table 6. In this table, for each initialization procedure, the “Avg. gap” row gives the overall average gap of the CPLEX results with the solutions obtained by utilizing the ILS algorithm on all the small and medium instances over the five runs. In addition, the best gap over five different runs is given in the row labeled by “Best gap”. Finally, the average computing time for all the small and medium instances over the five different random seeds is given in the row represented by “Avg. time”. According to the results, on average, all combinations have better performance with respect to CPLEX. Considering the best performance of the algorithm, best methods for generating initial solution are Sequential algorithm by using the S_2^j as the scoring

Table 5 Parameter tuning

Algorithm	Parameter	Tested values	Chosen value
ILS	<i>iter</i> 0	{110,120,130}	120
	<i>iter</i> 1	{250,200}	200
	θ	{0.7,0.5}	0.7
TS	<i>iter</i> 2	{110,120,130}	120
	<i>iter</i> 3	{300,400}	400
	<i>iter</i> 4	{20,30}	30
	<i>iter</i> 5	{50,60}	60
	τ	{5,8,10}	8
	θ	{0.7,0.5}	0.7
VNS	<i>iter</i> 6	{110,120,130}	120
	<i>iter</i> 7	{30,40,50}	40
	ρ_1	{0.2,0.3,0.4}	0.3
	ρ_{max}	{0.75,0.8,0.85}	0.8
	γ	{0.02,0.05}	0.02
	θ	{0.7,0.5}	0.7

Table 6 ILS results for small and medium instances of the TCMCRP

Initial solution type	Sequential (S_1^j)	Sequential (S_2^j)	Parallel I (S_1^j)	Parallel I (S_2^j)	Parallel II (S_1^j)	Parallel II (S_2^j)	Total
Avg. gap	-0.42	-0.44	-0.43	-0.25	-0.43	-0.36	-0.46
Best gap	-0.45	-0.46	-0.46	-0.28	-0.46	-0.42	-0.48
Avg. time	1.82	1.71	1.85	1.95	1.95	2.98	1.96

method, and Parallel I and Parallel II algorithms by using S_1^j as the scoring method. Finally, for each row of the table, the column labeled by “Total” gives the average of values over the corresponding row.

4.6 VNS results for small and medium instances

For small and medium instances, the results of applying the VNS algorithm are given in Table 7. In particular, each row of this table reports the same information as that already described for Table 6. Taking into account the average performance of the algorithm over five runs, Parallel II algorithm and Sequential algorithm with S_1^j as the scoring method have the best performance. In addition, Sequential method with S_1^j scoring method, Parallel I with S_1^j scoring method and Parallel II have the best performance, when considering the best performance of the algorithm. Based on the results reported in the last column (i.e., Total), on average, the ILS algorithm outperforms the VNS algorithm.

Table 7 VNS results for small and medium instances of the TCMCRP

Initial solution type	Sequential (S_1^J)	Sequential (S_2^J)	Parallel I (S_1^J)	Parallel I (S_2^J)	Parallel II (S_1^J)	Parallel II (S_2^J)	Total
Avg. gap	-0.34	-0.31	-0.30	-0.13	-0.34	-0.34	-0.29
Best gap	-0.41	-0.40	-0.41	-0.20	-0.41	-0.41	-0.41
Avg. time	0.13	0.13	0.13	0.12	0.13	0.13	0.13

Table 8 TS results for small and medium instances of the TCMCRP

Initial solution type	Sequential (S_1^J)	Sequential (S_2^J)	Parallel I (S_1^J)	Parallel I (S_2^J)	Parallel II (S_1^J)	Parallel II (S_2^J)	Total
Avg. gap	-0.43	-0.44	-0.45	-0.25	-0.46	-0.46	-0.49
Best gap	-0.46	-0.52	-0.46	-0.33	-0.52	-0.52	-0.52
Avg. time	2.09	1.75	1.90	1.88	1.87	1.80	2.12

4.7 TS results for small and medium instances

In this section, we analyze the results obtained by the TS algorithm for small and medium instances. According to the results reported in Table 8 and taking into account the average performance of the algorithm over 5 independent runs, Parallel II algorithm has the best performance. In addition, Sequential method with S_2^J as the scoring method and Parallel II are the best scenarios when considering the best performance of the algorithm. Finally, comparing the results obtained by all three heuristic algorithms indicates the superiority of the TS algorithm for the small and medium instances. In particular, when considering the average performance of the algorithms, the overall average gap is -0.49 for the TS algorithms, while this is -0.29 and -0.46 for the VNS and ILS algorithms, respectively.

4.8 Comparing the performance of the heuristic algorithms for large instances

In this section we provide the results of the CPLEX and heuristic algorithms over the set of large instances. In particular, according to the results reported in Tables 6, 7 and 8, for each of the heuristic algorithms the best scenario is chosen to run the experiments.

Each heuristic algorithm is run five times, and the average results are reported. The results show that for 38 instances in which CPLEX is able to reach a feasible solution, the proposed TS algorithm can improve the result by 2.64%, while this is 1.66% and 1.94% for the VNS and ILS algorithms, respectively. In addition, there are 133 instances for which TS has obtained the best solution, while VNS and ILS have achieved the best performance in 109 and 112 cases, respectively. Detailed results of the heuristic algorithms over the large size instances are provided in Appendix 1.

4.9 Computational results for the TCMCSP

As discussed in the definition of the problem, the TCMCSP is a special variant of our proposed problem. So, we have adopted our TS algorithm to be applied for this problem. For generating initial solutions, we have used Parallel II algorithm with S_1^j scoring method. The results are reported in Table 9, in which columns labeled by Wl and L_k have the same meaning as those already introduced in Table 1. Column labeled by “ $Best\ Obj.^E$ ” gives the best results achieved by the exact algorithm, proposed in Naji-Azimi and Salari (2014). For each instance “ $Avg.\ Obj.^H$ ” and “ $Best\ Obj.^H$ ” give, respectively, the best and the average performance of the proposed heuristic algorithm in Naji-Azimi and Salari (2014) over 5 different runs. Columns “ $Avg.\ Obj.^T$ ” and “ $Best\ Obj.^T$ ” give the average and the best results of the TS algorithm over five different runs, respectively. “ $Best\ Gap^{TE}$ ” column represents the gap between the best performance of TS algorithm (i.e., column $Best\ Obj.^T$) and the results reported in column “ $Best\ Obj.^E$ ”. Finally, the column “ $Best\ Gap^{TH}$ ” shows the gap between the best performance of our TS algorithm and the best performance of the heuristic algorithm proposed in Naji-Azimi and Salari (2014). As it is given in the last row of the table, on average, the TS algorithm improves the best known heuristic results in Naji-Azimi and Salari (2014) by 0.15%.

Taking into account the best performance of the TS algorithm, the results show that this algorithm is able to improve the solutions obtained by CPLEX by 1.27%. Average running time for the TS is 1.66 s which is a great improvement in comparison with 1680.24 s in CPLEX. We should mention that the average time of the proposed heuristic algorithm in Naji-Azimi and Salari (2014) is 1.54 s.

4.10 Computational results for the TOP

As already reported in Sect. 1, the TCMCRP is a generalization of the TOP. For this reason, in order to validate our proposed algorithm, we have modified the TS algorithm to be used for the TOP instances. In particular, we use 212 benchmark instances developed by Chao et al., which are divided into 4 datasets (Chao et al. 1996). In each dataset, the position and the profits of customers are the same for all instances, while the number of vehicles varies between 2 and 4. In addition, the travel limit is different for available instances.

We compare the performance of the TS with three heuristic algorithms, namely MA, PSOiA and PSOMA, and the results are reported in Table 10. In particular, the MA is a memetic algorithm proposed in (Bouly et al. 2010). PSOiA is a heuristic algorithm that uses the particle swarm optimization (PSO) to solve the problem (Dang et al. 2011). Finally, the PSOMA is a PSO-based memetic algorithm that combines the PSO and the memetic algorithms (Dang et al. 2013).

For each algorithm tested on an instance, the algorithm is run 10 times. Following this step, for each instance the best obtained solution (Z_{best}) by utilizing 4 different algorithms is used to analyze the effectiveness of the proposed TS algorithm and the results are reported in Table 10. In this table, for each instance set, the average gap

Table 9 Computational results for the TCMCSP instances

Instance	W	L_k	$BestObj^E$	$Avg.Obj^H$	$BestObj^H$	$Avg.Obj^T$	$BestObj^T$	$BestGap^E$	$BestGap^T$	$BestGap^{TH}$	Avg. time
berlin 52	36	547.23	1.00	1.00	1.00	1.00	1.00	0.00	0.00	0.00	0.00
berlin 52	36	820.84	5.00	5.00	5.00	5.00	5.00	0.00	0.00	0.00	0.02
berlin 52	36	1094.46	7.00	7.00	7.00	7.00	7.00	0.00	0.00	0.00	0.03
berlin 52	31	682.95	4.00	4.00	4.00	4.00	4.00	0.00	0.00	0.00	0.02
berlin 52	31	1024.42	6.00	6.00	6.00	6.00	6.00	0.00	0.00	0.00	0.09
berlin 52	31	1365.59	10.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00	0.14
berlin 52	26	795.29	6.00	6.00	6.00	6.00	6.00	0.00	0.00	0.00	0.11
berlin 52	26	1192.94	10.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00	0.11
berlin 52	26	1590.59	12.00	12.00	12.00	12.00	12.00	0.00	0.00	0.00	0.14
pr76	53	4784.19	5.00	5.00	5.00	5.00	5.00	0.00	0.00	0.00	0.02
pr76	53	7176.29	7.00	7.00	7.00	7.00	7.00	0.00	0.00	0.00	0.03
pr76	53	9568.35	7.00	7.00	7.00	7.00	7.00	0.00	0.00	0.00	0.11
pr76	45	7025.74	7.00	7.00	7.00	7.00	7.00	0.00	0.00	0.00	0.05
pr76	45	10,538.61	10.00	10.00	10.00	10.00	10.00	0.00	0.00	0.00	0.17
pr76	45	14,051.48	12.00	12.00	12.00	12.00	12.00	0.00	0.00	0.00	0.17
pr76	38	8960.28	7.00	7.00	7.00	7.00	7.00	0.00	0.00	0.00	0.14
pr76	38	13,440.42	11.00	11.00	11.00	11.00	11.00	0.00	0.00	0.00	0.28
pr76	38	17,920.56	14.00	14.00	14.00	14.00	14.00	0.00	0.00	0.00	0.53
krD100	70	2477.53	11.00	11.00	11.00	11.00	11.00	0.00	0.00	0.00	0.50
krD100	70	3716.29	15.00	15.00	15.00	15.00	15.00	0.00	0.00	0.00	0.25
krD100	70	4955.06	19.00	19.00	19.00	19.00	19.00	0.00	0.00	0.00	0.36
krD100	60	3330.43	14.00	14.00	14.00	14.00	14.00	0.00	0.00	0.00	0.61
krD100	60	4995.69	23.00	23.00	23.00	23.00	23.00	0.00	0.00	0.00	0.53

Table 9 continued

Instance	W	L_k	$BestObj.^E$	$Avg.Obj.^H$	$BestObj.^H$	$Avg.Obj.^T$	$BestObj.^T$	$BestGap^{TE}$	$BestGap^{TH}$	Avg. time
kroD100	60	6660.86	34.00	34.00	34.00	34.00	34.00	0.00	0.00	1.50
kroD100	50	4183.59	17.00	17.00	17.00	17.00	17.00	0.00	0.00	0.73
kroD100	50	6275.39	30.00	30.00	30.00	30.00	30.00	0.00	0.00	1.36
kroD100	50	6367.19	42.00	42.00	42.00	42.00	42.00	0.00	0.00	0.77
kroA150	105	3910.18	24.00	24.00	24.00	24.00	24.00	0.00	0.00	1.22
kroA150	105	5865.73	35.00	35.00	35.00	35.00	35.00	0.00	0.00	0.78
kroA150	105	7820.97	47.00	47.00	47.00	47.00	47.00	0.00	0.00	2.08
kroA150	90	5048.01	34.00	34.00	34.00	34.00	34.00	0.00	0.00	3.91
kroA150	90	7572.01	48.00	48.00	48.00	48.00	48.00	0.00	0.00	2.70
kroA150	90	10,096.02	62.00	62.00	62.00	62.00	62.00	0.00	0.00	1.88
kroA150	75	6274.96	38.00	39.00	39.00	38.50	39.00	-2.63	0.00	1.72
kroA150	75	9412.44	57.00	58.00	59.00	58.50	59.00	-3.51	0.00	3.80
kroA150	75	12,549.93	72.00	71.00	72.00	72.00	72.00	0.00	0.00	2.53
kroB150	105	3684.37	19.00	19.00	19.00	19.50	20.00	-5.26	-5.26	1.00
kroB150	105	5526.52	33.00	33.00	33.00	34.00	34.00	-3.03	-3.03	1.33
kroB150	105	7368.75	48.00	48.00	48.00	48.00	48.00	0.00	0.00	1.59
kroB150	90	4945.92	29.00	29.00	29.00	29.50	30.00	-3.45	-3.45	1.84
kroB150	90	7418.88	48.00	48.00	48.00	47.50	48.00	0.00	0.00	2.34
kroB150	90	9891.85	63.00	63.00	63.00	62.00	62.00	1.59	1.59	2.58
kroB150	75	6125.25	35.00	35.00	36.00	36.00	36.00	-2.86	0.00	1.33
kroB150	75	9189.37	55.00	55.00	55.00	56.00	56.00	-3.70	-1.82	2.72

Table 9 continued

Instance	IW1	L_k	$BestObj.^E$	Avg. $Obj.^H$	$BestObj.^H$	Avg. $Obj.^T$	$BestObj.^T$	$BestGap^{TE}$	$BestGap^{TH}$	Avg. time
kroB150	75	12,252.49	64.00	64.00	64.00	64.00	64.00	0.00	0.00	1.23
kroA200	140	4945.92	40.00	40.00	40.00	41.00	41.00	-2.50	-2.50	1.09
kroA200	140	7418.88	54.00	56.80	57.00	57.00	57.00	-5.56	0.00	1.66
kroA200	140	9891.89	75.00	78.00	78.00	76.00	76.00	-1.33	2.56	2.16
kroA200	120	6488.26	53.00	53.00	53.00	53.00	53.00	0.00	0.00	5.39
kroA200	120	9732.39	73.00	77.60	78.00	77.00	78.00	-6.85	0.00	5.31
kroA200	120	12,976.52	97.00	98.00	98.00	97.00	97.00	0.00	1.02	2.49
kroA200	100	8417.76	61.00	64.40	65.00	65.00	65.00	-6.56	0.00	4.69
kroA200	100	12,626.63	81.00	90.00	90.00	90.00	90.00	-11.11	0.00	4.22
kroA200	100	16,835.51	92.00	92.00	92.00	92.00	92.00	0.00	0.00	4.14
kroB200	140	5005.31	36.00	35.60	36.00	35.50	36.00	0.00	0.00	1.50
kroB200	140	7507.97	57.00	56.20	57.00	57.00	57.00	0.00	0.00	1.78
kroB200	140	10,010.63	76.00	75.20	76.00	76.00	76.00	0.00	0.00	3.09
kroB200	120	6669.35	52.00	53.00	53.00	53.00	53.00	-1.92	0.00	2.45
kroB200	120	10,004.02	79.00	79.60	80.00	79.00	79.00	0.00	1.25	3.91
kroB200	120	13,338.69	100.00	100.00	100.00	98.00	99.00	1.00	1.00	3.39
kroB200	100	8353.38	61.00	65.00	65.00	65.00	65.00	-6.56	0.00	6.06
kroB200	100	12,530.07	80.00	90.60	91.00	91.00	92.00	-15.00	-1.10	2.95
kroB200	100	16,706.76	93.00	94.00	94.00	94.00	94.00	-1.08	0.00	3.24
Average	38.35	39.00	39.11	39.01	39.11	-1.27	-0.15	1.66		

Table 10 Comparing the algorithms for the TOP instances

Instance set	p.4				p.5				p.6				p.7			
	MA10	PSOMA	PSOiA	TS	MA10	PSOMA	PSOiA	TS	MA10	PSOMA	PSOiA	TS	MA10	PSOMA	PSOiA	TS
Avg. gap	0.20	0.27	0.11	0.57	0.06	0.05	0.02	0.34	0.01	0.00	0.00	0.1	0.08	0.13	0.02	0.32
Best gap	0.03	0.03	0.00	0.27	0.04	0.01	0.00	0.20	0.00	0.00	0.00	0.1	0.01	0.02	0.00	0.21

Table 11 Efficiency of the local search procedures for the TCMCRP

Procedure type	Swap I	Swap II	Extraction–Insertion I	Extraction–Insertion II	Insertion	Deletion
Improvement percentage	34.07	19.66	37.50	16.28	0.72	1.46

and the best gap of different heuristics with respect to Z_{best} are reported in rows “Avg. gap” and “Best gap”, respectively.

The results show the efficiency of the TS algorithm. In particular, taking into account the average performance of the TS algorithm the worst results have been achieved in set p.4 for which the overall average gap with respect to the best solution is 0.57%. Considering the best performance of the TS algorithm, the results show that in the worst case the average gap is 0.27% for the set p.4. Detailed results of available algorithms are provided in Appendix 2.

4.11 The usefulness of the proposed local search procedures

In order to evaluate the effectiveness of the local search procedures which have been used in the structure of the proposed algorithms, we calculate the percentage of improvement made by each procedure as follows. In this regard, we consider the proportion of the iterations in which a specific procedure can improve the current solution, to the total number of the iterations that the procedure had been called. The obtained results are summarized in Table 11. As it can be concluded from this table, all the presented procedures are useful and the most effective procedure is Extraction–Insertion I, which has been successful in 37.50% of its execution.

4.12 Parameter analysis

In this section, we do some tests to analyze the impact of the parameters on the behavior of the TS algorithm. To do so, we fix all parameters and change one parameter at a time. Essentially, ρ and $iter2$ parameters have been selected. ρ is the percent of the facilities which has been removed from the solution in perturbation phase parameter (see Sect. 3.3.2). In our experiments $\rho \in \{0.6, 0.7, 0.8, 0.9\}$ and the impact of this parameter for the TCMCSP is given in Fig. 2. In this figure, the vertical column shows the overall gap between the TS and the best solutions proposed for the TCMCSP instances in Naji-Azimi and Salari (2014). In addition, Fig. 3 gives the same information for the $iter2$ parameter. We have selected five different scenarios, namely 90, 100, 110, 120 and 130, for running this test. Results show that the best result has been achieved when $iter2$ is equal to 120.

Figures 4 and 5 show the impact of $iter2$ and ρ parameters over the 135 instances of small and medium size for TCMCRP. Vertical axis shows the improvement in CPLEX’ results by applying the TS algorithm. The $iter2$ and ρ are set to 120 and 0.8, respectively.

Fig. 2 Analysis of parameter ρ for the TCMCSP instances

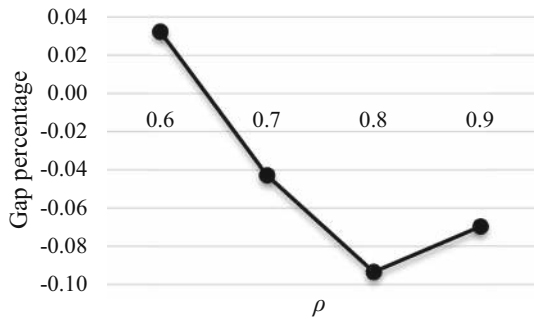


Fig. 3 Analysis of parameter *iter2* for the TCMCSP instances

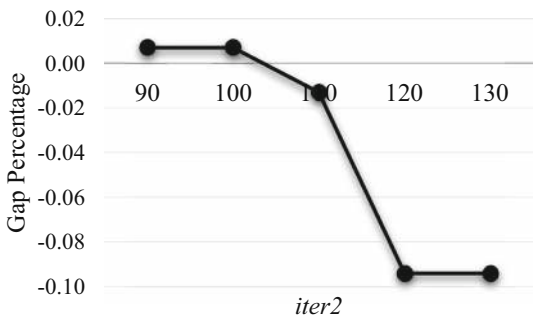


Fig. 4 Analysis of parameter *iter2* for the TCMCRP instances

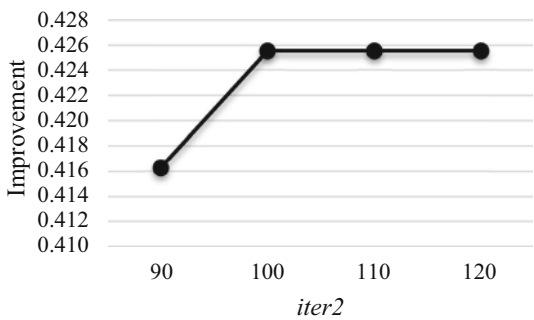


Fig. 5 Analysis of parameter ρ for the TCMCRP instances

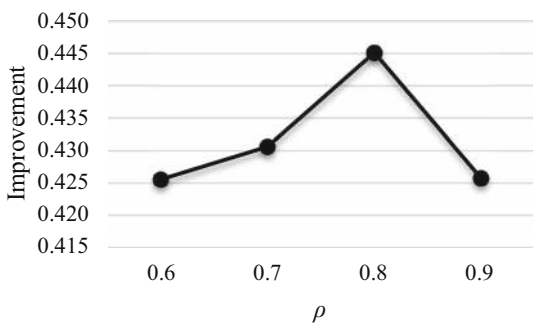


Fig. 6 Analysis of the relation between α and the total percentage of covered customers

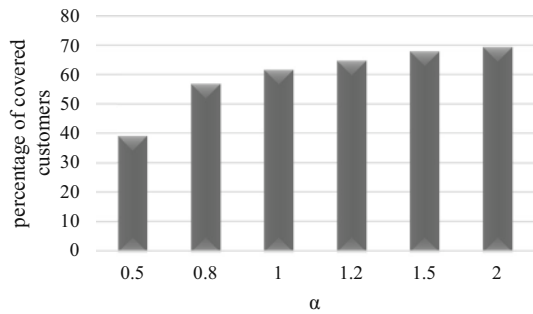
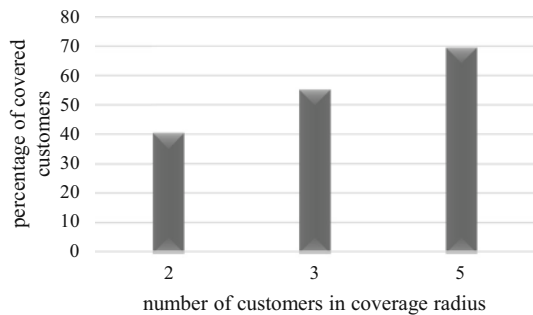


Fig. 7 Analysis of the relation between the number of the potential covered customers and the total percentage of covered customers



4.13 Analysis of model's parameters

In this section, we investigate the relation between L_k and the total percentage of covered customers. In particular, by increasing the corresponding value of α which concludes increasing L_k parameter, the percentage of covered customers increases as well. According to Fig. 6, the decision maker is able to select the best value for α by considering the number of customers whom he wants to cover.

Figure 7 shows the relation between the number of customers that can be covered by each facility and the percentage of customers that are covered by routed facilities. According to this figure, decision maker can determine the coverage radius of each facility by considering the distance between facilities and customers.

5 Conclusion

In this paper, the time-constrained maximal covering routing problem (TCMCRP) has been introduced, which is the combination of the covering salesman problem and the team orienteering problem. In this problem, a central depot, a set of facilities and a set of customers are given. The TCMCRP objective is to maximize the number of customers covered by facilities which are visited by a set of length constraint vehicles. We proposed a flow-based mathematical model to solve the model. Since the TCMCRP is a NP-hard problem, exact model is not effective for real size instances and heuristic algorithms including ILS, VNS and TS are presented for solving the problem. We tested the performance of these algorithms on 270 instances having 52–724 nodes

and dividing into small, medium and large size sets. The numerical results showed that on average all the proposed algorithms outperform CPLEX especially in large size instances. Regarding the performance of the heuristic algorithms, the tabu search algorithm has the best performance in comparison with the other two algorithms. In addition, to evaluate the effectiveness of the developed heuristic algorithms we have adapted our algorithms to be used for the time-constrained maximal covering salesman problem and the team orienteering problem. The results indicate that developed heuristic algorithms can achieve promising results for the similar problems. In particular, by applying the TS algorithm on the TCMCSP instances, the results show the superiority of the TS with respect to the best results available in the literature. The results also indicate that the TS algorithm performs efficiently when applying on the TOP benchmark instances.

For the future research, we propose to develop a variation of the problem in which we are given a threshold on the minimum percentage of customers that have to be covered while minimizing the total travel time. This problem happens in a situation that we have a limited amount of resources to be delivered to the customers within the shortest possible time.

Appendix 1

See Tables 12.

Table 12 Performance of the heuristic algorithms for large instances of the TCMGRP

Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)	Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)
1	40	40	40	40	51.69	42	-	65	65	65	108.47
2	37	37	37	37	59.82	43	-	58	58	58	79.15
3	-	37	37	37	62.11	44	-	61.5	62	62	81.19
4	41	41	41	41	55.04	45	-	63	63	63	98.35
5	40	40	40.6	42	58.21	46	75	75	75	75	45.77
6	-	39	39	39	56.61	47	72	72	72	72	64.68
7	-	40	40	40	59.54	48	58	71	71	75	67.84
8	-	39	39	39	56.30	49	73	73	73	73	59.97
9	-	39	39	39	57.91	50	73	73	73	73	65.24
10	36	36	36	36	41.86	51	-	71	71	71	66.42
11	41	41	41	41	43.83	52	75	75	75	75	56.92
12	33	36	37	37	46.99	53	-	72	72	72	65.39
13	39	39	39	39	44.38	54	-	67	67	67	64.76
14	43	43	43	43	46.87	55	-	27	27	27	134.17
15	-	44	44	44	51.37	56	-	29	29	29	123.58
16	40	40	40	40	38.80	57	-	28	28	28	148.83
17	-	40	40	40	44.35	58	-	30	30	30	235.21
18	-	38	38	38	41.83	59	-	31	31	31	149.97
19	43	43	43	43	32.93	60	-	28.5	30	30	168.48
20	38	43	43	43	25.76	61	-	30	30	30	151.52

Table 12 continued

Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)	Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)
21	43	36	36	36	33.36	62	-	30	30	30	172.51
22	41	40	39	41	29.23	63	-	30	30	30	143.45
23	37	39	39	37.8	27.91	64	27	27	27	27	113.68
24	37	40	37	40	38.62	65	-	27	27	27	111.39
25	41	43	43	41.8	31.07	66	-	30	30	30	137.35
26	40	39	39	40	31.56	67	-	28	28	28	116.41
27	-	40	39	41	25.39	68	-	29	29	29	104.40
28	58	58	58	58	119.81	69	-	27	25.5	27	100.56
Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)	Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)
29	54	54	54	54	121.73	70	-	26	26	26	118.94
30	-	58	58	58	149.92	71	-	30	30	30	96.77
31	58	58	57	58	100.45	72	-	29	29	29	143.71
32	-	58	58	58	115.17	73	27	27	27	27	76.56
33	-	56.8	56.6	57	114.68	74	26	26	26	26	58.93
34	-	57	57	57	123.46	75	-	30	30	30	62.97
35	-	58	58	59	122.05	76	30	30	30	30	76.56
36	-	58.6	58.8	60	114.36	77	-	27	27	27	55.06
37	62	63	63	63	84.07	78	-	28	28	28	60.30
38	64	64	64	64	103.12	79	19	21.5	22	26	84.69
39	-	65	65	65	88.53	80	-	29	29	29	58.97
40	63	63	63	63	80.20	81	-	28	28	28	66.77
41	-	61	61	61	103.00						

Table 12 continued

Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)
82	-	82	82	83	109	-	204	205	205	612.85
83	-	84	84	84	110	-	200	200	200	628.25
84	-	83	83	83	111	-	212.5	211	214	650.92
85	-	82	82	82	112	-	195	195	195	556.26
86	-	83	81	83	113	-	207	207	207	588.06
87	-	80	80	80	114	-	204	204	204	655.70
88	-	78	78	78	115	-	206	206	208	498.43
89	-	82	82	82	116	-	206	206	206	652.08
90	-	81	81	81	117	-	212.5	212.5	216	647.54
91	76	76	76	76	118	-	187	187	187	430.56
92	-	76	76	76	119	-	181.4	181.5	183	412.39
93	-	78	78	78	120	-	175	175	175	434.29
94	-	77	77	77	121	-	182.6	183.4	184	381.11
95	-	78	77	78	122	-	176	176	176	427.44
96	-	77	77	77	123	-	183	181.5	183	423.03
97	-	76	76	76	124	-	177	176.6	177	361.37
98	-	73	73	73	125	-	182	182	182	432.76
99	-	81	81	81	126	-	174	174	174	437.62
100	63	63	63	63	127	-	176	175	176	205.59

Table 12 continued

Instance	Obj. CPLEX	Avg. Obj. (ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. Obj. (TS)	Avg. time (TS)	Instance	Obj. CPLEX	Avg. Obj.(ILS)	Avg. Obj. (VNS)	Avg. Obj. (TS)	Avg. time (TS)
101	-	67	67	67	67	156.37	128	-	187	187	187	262.79
102	-	68	68	68	68	140.16	129	-	183	187	187	282.10
103	65	65	65	65	65	151.75	130	-	179	179	179	252.82
104	-	67	67	67	67	202.37	131	-	179	179	179	263.21
105	-	68	68	68	68	194.93	132	-	180	181	184	284.48
106	-	68	68	68	68	120.83	133	-	180	180	180	255.53
107	-	66	66	66	66	162.23	134	-	183	183	183	267.43
108	-	67	67	67	67	182.30	135	-	171.5	172	176	283.86

Bold values show best results

Appendix 2

See Tables 13, 14, 15, 16.

Table 13 Results for set 4 of the benchmark for the TOP

Instance	Z_{Best}	MA10			PSOMA			PSOiA			TS		
		Avg. gap	Best gap		Avg. gap	Best gap		Avg. gap	Best gap		Avg. gap	Best gap	
p.4.2.a	206	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.2.b	341	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.2.c	452	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.2.d	531	0.06	0.00	0.00	0.04	0.00	0.00	0.00	0.00	0.00	0.56	0.00	0.00
p.4.2.e	618	0.19	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.70	0.49	0.49
p.4.2.f	687	1.08	0.00	0.00	0.82	0.00	0.00	0.00	0.00	0.00	0.82	0.44	0.44
p.4.2.g	757	0.13	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	1.32	0.53	0.53
p.4.2.h	835	0.83	0.00	0.00	1.07	0.00	0.00	0.17	0.00	0.00	0.72	0.00	0.00
p.4.2.i	918	0.00	0.00	0.00	0.48	0.00	0.00	0.00	0.00	0.00	0.36	0.11	0.11
p.4.2.j	965	0.25	0.00	0.00	0.20	0.00	0.00	0.00	0.00	0.00	0.59	0.31	0.31
p.4.2.k	1022	0.13	0.00	0.00	0.18	0.00	0.00	0.10	0.00	0.00	0.78	0.39	0.39
p.4.2.l	1074	0.30	0.28	0.28	0.66	0.28	0.28	0.16	0.00	0.00	0.96	0.37	0.37
p.4.2.m	1132	0.23	0.00	0.00	0.19	0.00	0.00	0.12	0.00	0.00	0.77	0.35	0.35
p.4.2.n	1174	0.11	0.00	0.00	0.32	0.00	0.00	0.17	0.00	0.00	1.39	0.94	0.94
p.4.2.o	1218	0.16	0.00	0.00	0.52	0.00	0.00	0.58	0.00	0.00	0.41	0.25	0.25
p.4.2.p	1242	0.06	0.00	0.00	0.29	0.00	0.08	0.20	0.00	0.00	0.83	0.16	0.16
p.4.2.q	1268	0.29	0.08	0.08	0.27	0.08	0.08	0.14	0.00	0.00	0.79	0.47	0.47
p.4.2.r	1292	0.26	0.00	0.00	0.33	0.00	0.00	0.16	0.00	0.00	0.85	0.70	0.70
p.4.2.s	1304	0.17	0.00	0.00	0.15	0.00	0.00	0.02	0.00	0.00	0.46	0.15	0.15

Table 13 continued

Instance	Z_{best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.4.2.t	1306	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.3.b	38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.3.c	193	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.3.d	335	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.3.e	468	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.3.f	579	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.3.g	653	0.00	0.00	0.25	0.00	0.00	0.00	0.71	0.31
p.4.3.h	729	0.59	0.14	0.59	0.00	0.00	0.00	0.41	0.00
p.4.3.i	809	0.00	0.00	0.05	0.00	0.00	0.00	0.37	0.00
p.4.3.j	861	0.17	0.00	0.39	0.00	0.01	0.00	0.89	0.35
p.4.3.k	919	0.09	0.00	0.25	0.00	0.00	0.00	0.65	0.54
p.4.3.l	979	0.39	0.00	0.16	0.00	0.21	0.00	1.06	0.82
p.4.3.m	1063	0.48	0.00	0.43	0.00	0.09	0.00	1.47	1.32
p.4.3.n	1121	0.45	0.00	0.48	0.00	0.23	0.00	0.98	0.54
p.4.3.o	1172	0.26	0.00	0.20	0.00	0.00	0.00	1.82	0.94
p.4.3.p	1222	0.21	0.00	0.00	0.00	0.00	0.00	1.96	0.98
p.4.3.q	1253	0.24	0.00	0.22	0.00	0.06	0.00	1.01	0.48
p.4.3.r	1273	0.22	0.00	0.27	0.00	0.28	0.00	0.60	0.31
p.4.3.s	1295	0.09	0.00	0.27	0.00	0.42	0.00	0.05	0.00
p.4.3.t	1305	0.10	0.00	0.30	0.08	0.05	0.00	0.10	0.00
p.4.4.d	38	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.4.e	183	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.4.f	324	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 13 continued

Instance	Z_{best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.4.4.g	461	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.4.4.h	571	0.00	0.00	0.68	0.00	0.00	0.00	0.99	0.00
p.4.4.i	657	0.00	0.00	0.00	0.00	0.00	0.00	0.81	0.00
p.4.4.j	732	0.00	0.00	0.11	0.00	0.00	0.00	0.00	0.00
p.4.4.k	821	0.10	0.00	0.02	0.00	0.00	0.00	0.16	0.00
p.4.4.l	880	0.10	0.00	0.10	0.00	0.06	0.00	0.23	0.23
p.4.4.m	919	0.69	0.33	0.37	0.00	0.26	0.00	0.69	0.65
p.4.4.n	977	1.18	0.82	1.31	0.82	1.02	0.10	0.55	0.00
p.4.4.o	1061	0.31	0.00	0.89	0.00	0.09	0.00	0.69	0.57
p.4.4.p	1124	0.37	0.00	0.72	0.00	0.12	0.00	0.77	0.36
p.4.4.q	1161	0.00	0.00	0.14	0.00	0.00	0.00	0.75	0.43
p.4.4.r	1216	0.48	0.00	1.23	0.00	0.81	0.00	1.07	0.58
p.4.4.s	1260	0.25	0.00	0.23	0.08	0.20	0.00	0.61	0.08
p.4.4.t	1285	0.16	0.00	0.04	0.00	0.21	0.00	0.31	0.23
Average		0.20	0.03	0.27	0.03	0.11	0.00	0.57	0.27

Table 14 Results for set 5 of the benchmark for the TOP

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.5.2.b	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.c	50	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.d	80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.e	180	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.f	240	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.g	320	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.h	410	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.i	480	0.00	0.00	0.00	0.00	0.00	0.00	1.83	0.00
p.5.2.j	580	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.k	670	0.00	0.00	0.00	0.00	0.00	0.00	0.86	0.00
p.5.2.l	800	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.m	860	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.n	925	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.o	1020	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.p	1150	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.q	1195	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.r	1260	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.s	1340	1.12	0.75	0.78	0.00	0.00	0.00	2.43	2.24
p.5.2.t	1400	0.21	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.2.u	1460	0.00	0.00	0.00	0.00	0.00	0.00	0.86	0.00
p.5.2.v	1505	0.10	0.00	0.03	0.00	0.00	0.00	0.00	0.00

Table 14 continued

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.5.2.w	1565	0.32	0.32	0.32	0.32	0.16	0.00	0.32	0.32
p.5.2.x	1610	0.00	0.00	0.00	0.00	0.00	0.00	1.09	0.93
p.5.2.y	1645	0.00	0.00	0.00	0.00	0.00	0.00	0.30	0.00
p.5.2.z	1680	0.00	0.00	0.06	0.00	0.00	0.00	0.30	0.00
p.5.3.a	15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.b	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.c	60	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.d	95	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.e	110	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.f	185	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.h	260	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.i	335	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.j	470	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.k	495	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.l	595	0.00	0.00	0.00	0.00	0.00	0.00	0.84	0.00
p.5.3.m	650	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.n	755	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.o	870	0.00	0.00	0.00	0.00	0.00	0.00	2.01	1.15
p.5.3.p	990	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.q	1070	0.00	0.00	0.00	0.00	0.00	0.00	0.47	0.47
p.5.3.r	1125	0.00	0.00	0.00	0.00	0.00	0.00	0.22	0.00

Table 14 continued

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.5.3.s	1190	0.08	0.00	0.08	0.00	0.00	0.00	0.42	0.42
p.5.3.t	1260	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.3.u	1345	0.00	0.00	0.00	0.00	0.00	0.00	0.37	0.37
p.5.3.v	1425	0.00	0.00	0.00	0.00	0.00	0.00	0.35	0.00
p.5.3.w	1485	0.24	0.00	0.54	0.00	0.03	0.00	0.34	0.00
p.5.3.x	1555	0.48	0.00	0.55	0.00	0.19	0.00	0.80	0.00
p.5.3.y	1595	0.31	0.31	0.28	0.00	0.25	0.00	0.31	0.31
p.5.3.z	1635	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.c	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.d	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.e	20	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.f	80	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.g	140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.h	140	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.i	240	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.j	340	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.k	340	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.l	430	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 14 continued

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.5.4.m	555	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.n	620	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.o	690	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.p	765	0.65	0.65	0.59	0.00	0.59	0.00	0.65	0.65
p.5.4.q	860	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.r	960	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.s	1030	0.05	0.00	0.05	0.00	0.00	0.00	2.18	1.46
p.5.4.t	1160	0.00	0.00	0.00	0.00	0.00	0.00	3.88	3.45
p.5.4.u	1300	0.00	0.00	0.00	0.00	0.00	0.00	3.46	2.69
p.5.4.v	1320	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.w	1390	0.72	0.72	0.65	0.36	0.29	0.00	0.36	0.36
p.5.4.x	1450	0.00	0.00	0.14	0.00	0.00	0.00	0.34	0.00
p.5.4.y	1520	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.5.4.z	1620	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average		0.06	0.04	0.05	0.01	0.02	0.00	0.34	0.20

Table 15 Results for set 6 of the benchmark for the TOP

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.6.2.d	192	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p. 6.2.e	360	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p. 6.2.f	588	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.2.g	660	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.2.h	780	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.2.i	888	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.2.j	948	0.25	0.00	0.00	0.00	0.00	0.00	1.27	1.27
p.6.2.k	1032	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.2.l	1116	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.2.m	1188	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.2.n	1260	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.3.g	282	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.3.h	444	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.3.i	642	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 15 continued

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.6.3.j	828	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.3.k	894	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.3.l	1002	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.3.m	1080	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.3.n	1170	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.4.j	366	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.4.k	528	0.00	0.00	0.00	0.00	0.00	0.00	1.14	1.14
p.6.4.l	696	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.4.m	912	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.6.4.n	1068	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average		0.01	0.00	0.00	0.00	0.00	0.00	0.1	0.1

Table 16 Results for set 7 of the benchmark for the TOP

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.7.2.a	30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.b	64	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.c	101	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.d	190	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.e	290	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.f	387	0.16	0.00	0.16	0.00	0.00	0.00	0.00	0.00
p.7.2.g	459	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.h	521	0.08	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.i	580	0.10	0.00	0.16	0.00	0.02	0.00	0.34	0.34
p.7.2.j	646	0.00	0.00	0.11	0.00	0.00	0.00	0.15	0.00
p.7.2.k	705	0.18	0.00	0.11	0.00	0.04	0.00	0.78	0.71
p.7.2.l	767	0.00	0.00	0.00	0.00	0.00	0.00	0.65	0.00
p.7.2.m	827	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.2.n	888	0.01	0.00	0.05	0.00	0.00	0.00	1.46	1.24
p.7.2.o	945	0.00	0.00	0.00	0.00	0.00	0.00	1.16	1.16
p.7.2.p	1002	0.04	0.00	0.28	0.00	0.02	0.00	0.35	0.00
p.7.2.q	1044	0.02	0.00	0.46	0.00	0.03	0.00	0.96	0.96
p.7.2.r	1094	0.00	0.00	0.25	0.00	0.00	0.00	2.24	1.55
p.7.2.s	1136	0.00	0.00	0.13	0.00	0.00	0.00	1.28	0.70
p.7.2.t	1179	0.23	0.00	0.42	0.00	0.00	0.00	0.93	0.76

Table 16 continued

Instance	Z _{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.7.3.b	46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.3.c	79	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.3.d	117	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.3.e	175	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.3.f	247	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.3.g	344	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.3.h	425	0.05	0.00	0.19	0.00	0.00	0.00	0.00	0.00
p.7.3.i	487	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.3.j	564	0.07	0.00	0.23	0.00	0.00	0.00	0.62	0.18
p.7.3.k	633	0.05	0.00	0.16	0.00	0.00	0.00	0.16	0.16
p.7.3.l	684	0.28	0.00	0.29	0.15	0.00	0.00	0.44	0.44
p.7.3.m	762	0.00	0.00	0.14	0.00	0.00	0.00	0.59	0.13
p.7.3.n	820	0.00	0.00	0.29	0.00	0.00	0.00	0.61	0.49
p.7.3.o	874	0.34	0.00	0.17	0.00	0.00	0.00	0.57	0.46
p.7.3.p	929	0.32	0.00	0.45	0.22	0.11	0.00	0.54	0.43
p.7.3.q	987	0.00	0.00	0.65	0.00	0.00	0.00	1.06	0.71
p.7.3.r	1026	0.35	0.00	0.42	0.00	0.33	0.00	0.58	0.39
p.7.3.s	1081	0.12	0.00	0.14	0.00	0.00	0.00	0.37	0.19
p.7.3.t	1120	0.12	0.00	0.16	0.00	0.14	0.00	0.54	0.00
p.7.4.b	30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.c	46	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.d	79	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Table 16 continued

Instance	Z_{Best}	MA10		PSOMA		PSOiA		TS	
		Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap	Avg. gap	Best gap
p.7.4.e	123	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.f	164	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.g	217	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.h	285	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.i	366	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.j	462	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.k	520	0.31	0.00	0.35	0.00	0.35	0.00	0.00	0.00
p.7.4.l	590	0.36	0.00	0.27	0.00	0.00	0.00	0.00	0.00
p.7.4.m	646	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
p.7.4.n	730	0.55	0.55	0.56	0.55	0.16	0.00	0.55	0.55
p.7.4.o	781	0.20	0.00	0.22	0.00	0.08	0.00	0.38	0.38
p.7.4.p	846	0.27	0.00	0.54	0.00	0.00	0.00	0.00	0.00
p.7.4.q	909	0.22	0.00	0.22	0.00	0.03	0.00	0.88	0.33
p.7.4.r	970	0.00	0.00	0.00	0.00	0.00	0.00	0.21	0.00
p.7.4.s	1022	0.12	0.00	0.13	0.00	0.00	0.00	0.20	0.20
p.7.4.t	1077	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Average		0.08	0.01	0.13	0.02	0.02	0.00	0.32	0.21

References

- Aras N, Aksen D, Tekin MT (2011) Selective multi-depot vehicle routing problem with pricing. *Transp Res C Emerg* 19(5):866–884. <https://doi.org/10.1016/j.trc.2010.08.003>
- Archetti C, Feillet D, Hertz A, Speranza MG (2010) The undirected capacitated arc routing problem with profits. *Comput Oper Res* 37(11):1860–1869. <https://doi.org/10.1016/j.cor.2009.05.005>
- Bae H, Moon I (2016) Multi-depot vehicle routing problem with time windows considering delivery and installation vehicles. *Appl Math Model* 40(13):6536–6549. <https://doi.org/10.1016/j.apm.2016.01.059>
- Baldacci R, Dell'Amico M, González JS (2007) The capacitated m-ring-star problem. *Oper Res* 55(6):1147–1162. <https://doi.org/10.1287/opre.1070.0432>
- Bock A, Sanità L (2015) The capacitated orienteering problem. *Discrete Appl Math* 195:31–42. <https://doi.org/10.1016/j.dam.2014.10.001>
- Bouly H, Dang DC, Moukrim A (2010) A memetic algorithm for the team orienteering problem. *Q J Oper Res* 4 or 8(1):49–70. <https://doi.org/10.1007/s10288-008-0094-4>
- Buhrkal K, Larsen A, Ropke S (2012) The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia Soc Behav Sci* 39:241–254. <https://doi.org/10.1016/j.sbspro.2012.03.105>
- Butt SE, Cavalier TM (1994) A heuristic for the multiple tour maximum collection problem. *Comput Oper Res* 21(1):101–111. [https://doi.org/10.1016/0305-0548\(94\)90065-5](https://doi.org/10.1016/0305-0548(94)90065-5)
- Chao IM, Golden B, Wasil E (1996) The team orienteering problem. *Eur J Oper Res* 88:464–474. [https://doi.org/10.1016/0377-2217\(94\)00289-4](https://doi.org/10.1016/0377-2217(94)00289-4)
- Church R, Velle CR (1974) The maximal covering location problem. *Pap Reg Sci*. 32:101–1181. <https://doi.org/10.1111/j.1435-5597.1974.tb00902.x>
- Current JR, Schilling DA (1989) The covering salesman problem. *Transp Sci* 23:208–213. <https://doi.org/10.1287/trsc.23.3.208>
- Current JR, Schilling DA (1994) The median tour and maximal covering tour problems: formulations and heuristics. *Eur J Oper Res* 73(1):114–126. [https://doi.org/10.1016/0377-2217\(94\)90149-X](https://doi.org/10.1016/0377-2217(94)90149-X)
- Dang DC, Guibadj RN, Moukrim A (2011) A pso-based memetic algorithm for the team orienteering problem. In: *European conference on the applications of evolutionary computation*. Springer, Berlin, pp 471–480. https://doi.org/10.1007/978-3-642-20520-0_48
- Dang DC, Guibadj RN, Moukrim A (2013) An effective PSO-inspired algorithm for the team orienteering problem. *Eur J Oper Res* 229(2):332–344. <https://doi.org/10.1016/j.ejor.2013.02.049>
- Dantzig GB, Ramser JH (1959) The truck dispatching problem. *Manag Sci* 6(1):80–91. <https://doi.org/10.1287/mnsc.6.1.80>
- Dell'Amico M, Righini G, Salani M (2006) A branch-and-price approach to the vehicle routing problem with simultaneous distribution and collection. *Transp Sci* 40(2):235–247. <https://doi.org/10.1287/trsc.1050.0118>
- Faulin J, Juan A, Lera F, Grasman S (2011) Solving the capacitated vehicle routing problem with environmental criteria based on real estimations in road transportation: a case study. *Procedia Soc Behav Sci* 20:323–334. <https://doi.org/10.1016/j.sbspro.2011.08.038>
- Gendreau M, Laporte G, Semet F (1997) The covering tour problem. *Oper Res* 45:568–576. <https://doi.org/10.1287/opre.45.4.568>
- Glover F (1989) Tabu search-part I. *ORSA J Comput* 1(3):190–206. <https://doi.org/10.1287/ijoc.1.3.190>
- Golden BL, Raghavan S, Wasil EA (2008) *The vehicle routing problem: latest advances and new challenges*. Springer, New York
- Golden BL, Naji-Azimi Z, Raghavan S, Salari M, Toth P (2012) The generalized covering salesman problem. *INFORMS J Comput* 24(4):534–553. <https://doi.org/10.1287/ijoc.1110.0480>
- Gunawan A, Lau HC, Lu K (2015) An iterated local search algorithm for solving the orienteering problem with time windows. In: *European conference on evolutionary computation in combinatorial optimization*. pp 61–73. https://doi.org/10.1007/978-3-319-16468-7_6
- Gunawan A, Lau HC, Vansteenwegen P (2016) Orienteering problem: a survey of recent variants, solution approaches and applications. *Eur J Oper Res* 255(2):315–332. <https://doi.org/10.1016/j.ejor.2016.04.059>
- Hiermann G, Puchinger J, Ropke S, Hartl RF (2016) The electric fleet size and mix vehicle routing problem with time windows and recharging stations. *Eur J Oper Res* 252(3):995–1018. <https://doi.org/10.1016/j.ejor.2016.01.038>

- Hosseinabadi AAR, Rostami NSH, Kardgar M, Mirkamali S, Abraham A (2017) A new efficient approach for solving the capacitated vehicle routing problem using the gravitational emulation local search algorithm. *Appl Math Model* 49:663–679. <https://doi.org/10.1016/j.apm.2017.02.042>
<https://www.cdc.gov/outbreaks/index.html>. Accessed 06 Nov 2018
<https://ecdc.europa.eu/en/threats-and-outbreaks>. Accessed 07 Oct 2018
<http://www.who.int/csr/don/archive/year/2018/en/>. Accessed 07 June 2018
- Jabali O, Leus R, Van Woensel T, De Kok T (2015) Self-imposed time windows in vehicle routing problems. *OR Spect* 37(2):331–352. <https://doi.org/10.1007/s00291-013-0348-1>
- Kara I (2011) Arc based integer programming formulations for the distance constrained vehicle routing problem. In: 2011 3rd IEEE international symposium on logistics and industrial informatics (LINDI). pp 33–38. <https://doi.org/10.1109/LINDI.2011.6031159>
- Karaoglan I, Altıparmak F, Kara I, Dengiz B (2009) Vehicle routing problem with simultaneous pickup and delivery: mixed integer programming formulations and comparative analyses. *Res Pap*. http://w3.gazi.edu.tr/fulyaal/Papers/VRPSD_MIPFormulations.pdf
- Kergosien Y, Lenté C, Billaut JC, Perrin S (2013) Metaheuristic algorithms for solving two interconnected vehicle routing problems in a hospital complex. *Comput Oper Res* 40(10):2508–2518. <https://doi.org/10.1016/j.cor.2013.01.009>
- Labbé M, Laporte G, Martín IR, González JJS (2004) The ring star problem: polyhedral analysis and exact algorithm. *Networks* 43(3):177–189. <https://doi.org/10.1002/net.10114>
- Lalla-Ruiz E, Expósito-Izquierdo C, Taheripour S, Voß S (2016) An improved formulation for the multi-depot open vehicle routing problem. *OR Spect* 38(1):175–187. <https://doi.org/10.1007/s00291-015-0408-9>
- Laport G, Martello S (1990) The selective travelling salesman problem. *Discrete Appl Math* 26(2):193–207. [https://doi.org/10.1016/0166-218X\(90\)90100-Q](https://doi.org/10.1016/0166-218X(90)90100-Q)
- Lee Y, Chiu SY, Ryan J (1996) A branch and cut algorithm for a Steiner tree-star problem. *INFORMS J Comput* 8(3):194–201. <https://doi.org/10.1287/ijoc.8.3.194>
- Lourenço HR, Martin OC, Stutzle T (2003) Iterated local search. *Int Ser Oper Res Manag Sci* 146:321–354
- Mladenović N, Hansen P (1997) Variable neighborhood search. *Comput Oper Res* 24(11):1097–1100. [https://doi.org/10.1016/S0305-0548\(97\)00031-2](https://doi.org/10.1016/S0305-0548(97)00031-2)
- Montané FAT, Galvão RD (2006) A tabu search algorithm for the vehicle routing problem with simultaneous pick-up and delivery service. *Comput Oper Res* 33(3):595–619. <https://doi.org/10.1016/j.cor.2004.07.009>
- Naji-Azimi Z, Salari M (2014) The time constrained maximal covering salesman problem. *Appl Math Model* 38(15):3945–3957. <https://doi.org/10.1016/j.apm.2014.01.001>
- Pirkul H, Schilling DA (1991) The maximal covering location problem with capacities on total workload. *Manag Sci* 37(2):233–248. <https://doi.org/10.1287/mnsc.37.2.233>
- Pisinger D, Ropke S (2007) A general heuristic for vehicle routing problems. *Comput Oper Res* 34(8):2403–2435. <https://doi.org/10.1016/j.cor.2005.09.012>
- Potvin JY, Rousseau JM (1993) A parallel route building algorithm for the vehicle routing and scheduling problem with time windows. *Eur J Oper Res* 66(3):331–340. [https://doi.org/10.1016/0377-2217\(93\)90221-8](https://doi.org/10.1016/0377-2217(93)90221-8)
- Reinelt G (1991) A traveling salesman problem library. *ORSA J Comput* 3(4):376–384. <https://doi.org/10.1287/ijoc.3.4.376>
- Solomon MM (1987) Algorithms for the vehicle routing and scheduling problems with time window constraints. *Oper Res* 35(2):254–265. <https://doi.org/10.1287/opre.35.2.254>
- Sörensen K, Schittekat P (2013) Statistical analysis of distance-based path relinking for the capacitated vehicle routing problem. *Comput Oper Res* 40(12):3197–3205. <https://doi.org/10.1016/j.cor.2013.02.005>
- Souffriau W, Vansteenwegen P, Vanden Berghe G, Van Oudheusden D (2013) The multiconstraint team orienteering problem with multiple time windows. *Transp Sci* 47(1):53–63. <https://doi.org/10.1287/trsc.1110.0377>
- Sundar K, Rathinam S (2017) Multiple depot ring star problem: a polyhedral study and an exact algorithm. *J Glob Optim*. 67(3):527–551. <https://doi.org/10.1007/s10898-016-0431-7>
- Toth P, Vigo D (2001) The vehicle routing problem. Society for Industrial and Applied Mathematics, Philadelphia
- Tsiligirides T (1984) Heuristic methods applied to orienteering. *J Oper Res Soc* 35(9):797–809. <https://doi.org/10.2307/2582629>

- Vansteenwegen P, Souffriau W, Vanden Berghe G, Van Oudheusden D (2009) Iterated local search for the team orienteering problem with time windows. *Comput Oper Res* 36(12):3281–3290. <https://doi.org/10.1016/j.cor.2009.03.008>
- Vansteenwegen P, Souffriau W, Van Oudheusden D (2011) The orienteering problem: a survey. *Eur J Oper Res* 209(1):1–10. <https://doi.org/10.1016/j.ejor.2010.03.045>
- Wasner M, Zäpfel G (2004) An integrated multi-depot hub-location vehicle routing model for network planning of parcel service. *Int J Prod Econ* 90(3):403–419. <https://doi.org/10.1016/j.ijpe.2003.12.002>
- Wilcoxon F (1945) Individual comparisons by ranking methods. *Biom Bull* 1(6):80–83. <https://doi.org/10.2307/3001968>
- Zachariadis EE, Kiranoudis CT (2011) Local search for the undirected capacitated arc routing problem with profits. *Eur J Oper Res* 210(2):358–367. <https://doi.org/10.1016/j.ejor.2010.09.039>