



# A variable space search heuristic for the Capacitated Team Orienteering Problem

Asma Ben-Said<sup>1</sup> · Racha El-Hajj<sup>1,2</sup> · Aziz Moukrim<sup>1</sup>

Received: 7 January 2018 / Revised: 12 September 2018 / Accepted: 17 September 2018 /

Published online: 28 September 2018

© Springer Science+Business Media, LLC, part of Springer Nature 2018

## Abstract

The Capacitated Team Orienteering Problem (CTOP) is a variant of the well-known Team Orienteering Problem where additional capacity limitation constraints are considered for each vehicle. Solving CTOP consists of organizing a set of routes that maximize the total profit collected from the served customers while taking into consideration the capacity and travel time limitation for each vehicle. In this paper, we propose a variable space search heuristic to solve CTOP. Our algorithm alternates between two search spaces: the *giant tour* and *routes* search spaces. We develop a hybrid heuristic as a framework for our algorithm composed of a combination between Greedy Randomized Adaptive Search Procedure and Evolutionary Local Search. Several local search techniques were developed in each search space to improve the quality of the solutions and the giant tours. A dedicated optimal split procedure and a concatenation technique are performed to ensure the link between the search spaces. This approach shows its high performance on the benchmark of CTOP, and proves its competitiveness in comparison to the other heuristic methods available in the literature as it yields to strict improvements with small computational time.

**Keywords** Vehicle routing · Capacitated Team Orienteering Problem · Variable space search · Local search

---

✉ Asma Ben-Said  
asma.ben-said@hds.utc.fr

Racha El-Hajj  
racha.elhajj@hds.utc.fr

Aziz Moukrim  
aziz.moukrim@hds.utc.fr

<sup>1</sup> CNRS, Heudiasyc UMR 7253, CS 60 319, Université de technologie de Compiègne, Sorbonne universités, 60 203 Compiègne Cedex, France

<sup>2</sup> Faculté de génie, Université Libanaise, Campus Hadath, Beyrouth, Lebanon

## 1 Introduction

In vehicle routing problems with profits, each customer has its own profit and due to some resource limitation constraints, it is often not possible to serve all customers. In general, the aim of solving this category of problems is to determine which customers must be served and how to design the vehicle routes so that the total collected profit is maximized while adhering to resource limitation. Based on the literature, several variants of the vehicle routing problem with profits appeared to depend on the nature of the resource limitation.

The Profitable Tour Problem (PTP) is one of the variant where only one vehicle is considered (Feillet et al. 2005). The objective of solving PTP is to maximize the difference between the collected profit and the total traveling cost. This cost is referred to as the total distance traveled or the operational time consumed. The Capacitated-PTP is the multiple vehicles extension of the PTP where a capacity constraint is imposed on each vehicle (Jepsen et al. 2014). In the Prize Collecting Traveling Salesman Problem (PCTSP), a single vehicle is available and its objective is to minimize the total traveling cost while visiting enough customers to ensure a predefined profit (Feillet et al. 2005). Alternatively, the Orienteering Problem (OP) consists of designing a single route in order to maximize the collected profit that respects a specified travel time limit (Chao et al. 1996). As an extension of OP, by considering multiple vehicles for the service of the customers, the well-known Team Orienteering Problem (TOP) (Kim et al. 2013) was initially introduced by (Butt and Cavalier 1994) as the Multiple Tour Maximum Collection Problem. The aim of solving TOP is to find a set of vehicles routes that maximizes the total collected profit while abiding to the travel time limit for each vehicle.

In this paper, we focus on the Capacitated Team Orienteering Problem (CTOP), a new variant of the Team Orienteering Problem (TOP). CTOP was recently introduced by Archetti et al. (2009) and considered more operational constraints by imposing capacity restrictions to vehicles. Similarly to the Team Orienteering Problem, CTOP is a selective routing problem, where some of the customers cannot be visited due to some resources limitations. In this problem, a set of customers with known profits and demands are to be served by a fleet of vehicles. Each customer can be visited at most once by a single vehicle. Vehicles are initially located at the departure depot and are required to finish at the arrival depot without exceeding a predefined travel time. Moreover, the total demand of customers visited in any route must fit the vehicle capacity. The problem is to design a set of vehicle routes that maximizes the total collected profit and satisfies all resources limitations.

The Capacitated Team Orienteering Problem is related to many real life applications. It is mainly used by industries to model and solve product delivery tasks, aiming at improving products distribution to customers. Sometimes, it is more convenient to outsource the distribution task to other specified companies. The aim of each transportation company is to identify convenient customers to add to their pre-assigned routes while respecting resources limitations, i.e. number of vehicles, capacity and travel time limit of each vehicle. Therefore, the problem can be formulated as a capacitated team orienteering problem where the visited customers have to be selected and assigned to the available vehicles so that the total collected profit is maximized

while respecting all the operational restrictions. Despite its practical importance, few researchers had previously addressed CTOP in the literature so far. Archetti et al. (2009) proposed an exact method, which is based on the branch-and-price algorithm developed by Boussier et al. (2007) for TOP. The column generation algorithm deals with a set covering problem, and the dynamic programming algorithm is used to solve the elementary shortest path problems that model the subproblems. Furthermore, Archetti et al. (2009) proposed two Tabu Searches (TSF and TSA) and a variable neighborhood search (VNS) algorithm for CTOP. The main idea is based on assigning all customers to various routes. Customers are iteratively shifted between the routes until the most profitable routes are chosen to build the solution. In addition to feasible solutions search space, they explore admissible solutions search space where the most profitable routes of a solution can violate length and capacity constraints. Finally, repair and jump mechanisms are used. Both Tabu Search methods explore a small region in the solution space, unlike the VNS method where several moves are performed to explore a larger region.

Archetti et al. (2013) proposed an exact method based on the branch-and-price algorithm to solve CTOP. Later, Tarantilis et al. (2013) designed a two-phase heuristic method called Bi-Level Filter-and-Fan method (BiF&F). In their algorithm, two heuristics are alternatively applied: a Tabu Search procedure and a heuristic decomposition scheme. The search is performed according to two levels. The upper level deals with a profit oriented search while the lower one aims to minimize the travel time. Recently, Luo et al. (2013) proposed an efficient heuristic method called ADaptive Ejection Pool with Toggle-Rule Diversification (ADEPT-RD). In their method, at each iteration, only one customer is inserted in the solution from a maintained priority queue of unvisited customers. During the search, the priority queue is updated with an adaptive mechanism and reinitialized with different sorting rules to avoid local optima. The method iteratively inserts customers having the highest valuations from the priority queue into the solution. Then, a local search technique is performed to restore the feasibility of the solution. Consequently, their approach integrates a post-optimization heuristic to improve the quality of the best solution obtained.

This paper presents a Variable Space Search (VSS) heuristic that alternates between the *routes* and the *giant tour* search spaces, under the framework of a Greedy Randomized Adaptive Search Procedure (GRASP) combined with an Evolutionary Local Search (ELS). This algorithm proved successful as a result of alternation between the two search spaces, and to the improvements performed on the *routes* and the *giant tours* by the local searches applied in both spaces. Our main contributions are summarized as follows.

- One of the key benefits of our proposed algorithm is the fact that it operates in two search spaces, the *giant tour* search space and the feasible CTOP solution represented by the *routes* search space, where several improvements were integrated in both spaces using different types of local searches. A powerful optimal split procedure is developed to extract the optimal feasible CTOP solution from the *giant tour*.
- Our algorithm makes use of an efficient framework combining the Greedy Randomized Adaptive Search Procedure (GRASP) and the Evolutionary Local Search

(ELS). Our approach synthesizes the positive aspects of all the involved techniques. GRASP explores different regions from the search spaces and ELS extensively explores one region before moving to the next one. An initial solution of good quality is given to GRASP using an adaptive Iterative Destruction/Construction Heuristic (IDCH).

- Several adaptive local searches are performed to improve the quality of the solutions, as well as the quality of the giant tours. These local searches allow ELS to avoid local optima while exploring a certain region. Besides the constructive/destructive heuristic applied on the solution of CTOP, a Tabu Search and a simulated annealing (SA) procedures were also involved to further improve the quality of the solutions. Therefore, we proposed two versions of our VSS algorithm, VSS-Tabu that uses the Tabu Search as one of its local searches, and VSS-SA that integrates the simulated annealing instead.
- Several experiments were conducted on the benchmark of CTOP proposed by Archetti et al. (2009) and Tarantilis et al. (2013). Our two versions of VSS algorithm are able to outperform the results found in the literature in terms of gap and computational time. VSS-Tabu had found 74 new best solutions with a gap of  $-0.242$ , while VSS-SA had reached a gap of  $-0.227$  with 77 new improvements.

The remainder of this paper is set as follows. In Sect. 2, we provide a formal description of CTOP. Our VSS algorithm is described in Sect. 3 where the initialization algorithm, the alternation between the two search spaces and several components are explained in details. Computational experiments are reported in Sect. 4, followed by conclusions and further development.

## 2 Problem definition

Given a set  $V^- = \{1, \dots, n\}$  of  $n$  customers and a fleet of  $m$  vehicles located initially at the depot 0, CTOP can be modeled with a complete graph  $G = (V, E)$ , where  $V = V^- \cup \{0\}$  is the set of vertices and  $E = \{(i, j) | i, j \in V\}$  is the set of edges. A travel time  $C_{ij}$  is associated with each edge  $(i, j) \in E$  and respects the triangle inequalities. In addition, a profit  $p_i$  is associated with each customer  $i \in V^-$  and is collected once its demand  $d_i$  is delivered. Furthermore, each customer  $i$  is assigned a service time  $s_i$ . Having that split deliveries are not allowed, a customer can be visited at most once. For each vehicle, a maximum capacity  $Q$  and a maximum traveling time limit  $T_{max}$  are assigned to its trip from the starting to the ending depot. Due to resource limitations, some customers might remain unserved. Therefore, solving CTOP consists of selecting the subset of customers to be visited and organizing the vehicles routes in order to maximize the total collected profit while accounting for all resource limitation constraints related to the capacity of vehicles and to the maximum traveling time.

A feasible solution  $S$  of CTOP is composed of  $m$  routes  $r_k, k = 1, \dots, m$ . We denote  $r_k$  the ordered list of  $n_k$  customers visited by vehicle  $k$ . A feasible route  $r_k$  must comply with the following constraints:

- Each route should start and end at the depot:  $r_k = (0, r_k[1], \dots, r_k[n_k], 0)$

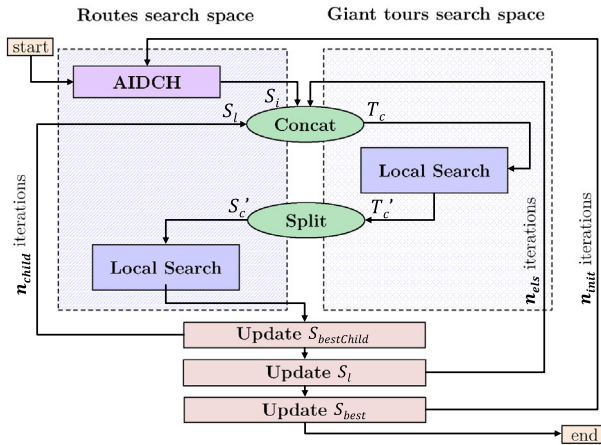


Fig. 1 VSS general structure

- The time needed to visit the customers of each route should not exceed the maximum travel time ( $T$  - constraint for brevity):

$$T(r_k) = C_0 r_k[1] + \sum_{i=1}^{n_k-1} (C_{r_k[i] r_k[i+1]} + s_{r_k[i]}) + C_{r_k[n_k] 0} + s_{r_k[n_k]} \leq T_{max}$$

- The total demand of customers visited in each route is upper bounded by the

$$\text{vehicle capacity (} C \text{ - constraint for brevity): } D(r_k) = \sum_{i=1}^{n_k} d_i \leq Q$$

### 3 VSS heuristic for CTOP

In this section, our global VSS algorithm is described. This algorithm ensures the alternation between the two search spaces under the framework of GRASP-ELS, where adaptive local searches are applied in both spaces.

#### 3.1 Global scheme of the GRASP-ELS framework

The efficiency of our algorithm is the alternation between two search spaces, as shown in Fig. 1: the first one for CTOP solutions, each encoded as a set of routes that respect all the resource limitations and the second one for solutions encoded as giant tours defined as a sequence of the  $n$  customers. This principle is shown to be successful on many combinatorial optimization problem (Hertz et al. 2008), and more precisely on vehicle routing problems (Duhamel et al. 2010, 2011; Lacomme et al. 2013). In the GRASP-ELS framework, ELS is applied as local search on each initial solution given by GRASP, while SA and Tabu Search are applied in ELS to decide which region to explore. We note that during the transition between the two search spaces, we ensure that there is no loss of information concerning the quality of the solution.

A CTOP solution is converted into a giant tour using the *Concat* procedure, where the depot is first removed from each route. Resulting routes are then concatenated

with a random order and unrouted customers are then inserted randomly in between the routes. The reverse operation is done by a dedicated *Split* procedure described in Sect. 3.4.1. This procedure is used to convert giant tour into an optimal CTOP solution with respect the customers order in the giant tour. Our VSS algorithm is summarized in Algorithm 1.

---

**Algorithm 1:** General structure of VSS
 

---

**Parameters:**  $n_{init}$  : number of initial solutions  
 $n_{els}$  : number of ELS levels  
 $n_{child}$  : number of generated child solutions

```

1  $S_{best} \leftarrow \emptyset; S_i \leftarrow \emptyset;$ 
2 for  $i \leftarrow 1$  to  $n_{init}$  do
3   AIDCH ( $S_i$ );
4    $S_l \leftarrow S_i;$ 
5   for  $l \leftarrow 1$  to  $n_{els}$  do
6      $S_{bestChild} \leftarrow \emptyset;$ 
7     for  $c \leftarrow 1$  to  $n_{child}$  do
8        $T_c \leftarrow \text{Concat} (S_l);$ 
9        $T'_c \leftarrow \text{Local Search} (T_c);$ 
10       $S'_c \leftarrow \text{Split} (T'_c);$ 
11       $S''_c \leftarrow \text{Local Search} (S'_c);$ 
12      if ( $\mathbb{P}(S''_c) \geq \mathbb{P}(S_{bestChild})$ ) then
13         $S_{bestChild} \leftarrow S''_c;$ 
14      if ( $\mathbb{P}(S_{bestChild}) \geq \mathbb{P}(S_l)$ ) then
15         $S_l \leftarrow S_{bestChild};$ 
16      if ( $\mathbb{P}(S_l) \geq \mathbb{P}(S_{best})$ ) then
17         $S_{best} \leftarrow S_l;$ 
18 return  $S_{best};$ 

```

---

In this algorithm, the outer loop controls the structure of GRASP, where  $n_{init}$  solutions are generated and considered as starting points for ELS. These CTOP initial solutions  $S_i$  are obtained using an Adaptive Iterative Destruction/Construction Heuristic (AIDCH), described in Sect. 3.2.1. The middle loop corresponds to the ELS loop where  $n_{els}$  iterations are performed. The best solution obtained after  $l$  iterations is recorded in  $S_l$ . During the inner loop,  $n_{child}$  parallel child solutions are generated by making  $n_{child}$  copies from  $S_l$ . Each child  $S_c$  is converted into a giant tour  $T_c$  using the *Concat* procedure. Each giant tour is evaluated with the *Split* procedure and is improved using the *local search giant tours space* procedure. The resulting giant tour  $T'_c$  is evaluated with the split procedure to obtain the CTOP solution  $S'_c$ . To further explore the neighborhood of the new solution, a local search in *routes search space* is performed on  $S'_c$  either by applying a Simulated Annealing procedure or a Tabu Search as described in Sect. 3.3. Finally, the resulting solution  $S''_c$  updates the best child  $S_{bestChild}$ . The incumbent solution is retained during the whole algorithm to be returned at the end of the algorithm as the best solution found  $S^*$ .

### 3.2 Routes search space

The first search space explored by our VSS algorithm is the *routes* search space, in which only feasible CTOP solutions that respect all the imposed constraints are considered. GRASP starts in this search space with an initial solution of good quality generated by AIDCH. After performing several improvements to this solution in the *giant tour* search space, a split procedure is applied on the resulting giant tour to extract from it the optimal CTOP solution on which other local searches are then applied in the *routes* search space as the SA procedure and the Tabu Search.

#### 3.2.1 Adaptive iterative destruction/construction heuristic

To generate initial solutions for our GRASP, we propose an Adaptive Iterative Destruction/Construction Heuristic (AIDCH) composed of two main procedures: an adaptive construction algorithm and an adaptive diversification procedure. In our adaptive heuristic, several parameters are changed according to the solution process.

The core component of the adaptive construction procedure is the Best Insertion Algorithm (BIA). It starts first with an incomplete solution  $S$  and a subset  $U$  of unrouted customers, then begins to insert customers from  $U$  in the solution one by one. The process ends when no possible insertions exist or no customers remain unserved. At each iteration a new customer is chosen to be inserted in the solution if its insertion preserves the capacity and the travel time restrictions of all the routes and has the lowest insertion cost. For this reason, we evaluate at each iteration, the insertion of each customer  $c$  from  $U$  in every position  $p$  in  $S$ . In order to determine the best customer to be inserted in the best position, a ratio is calculated for each possible insertion:

$$Ratio_{cp} = \frac{\left(\frac{\Delta L_{cp}}{L_{max}}\right)^{\beta} * \left(\frac{d_c}{Q}\right)^{\gamma}}{\left(\frac{p_c}{P_{max}}\right)^{\alpha}} \quad (1)$$

$\Delta L_{cp}$  is defined as the difference of travel time resulting from inserting customer  $c$  in position  $p$  and  $P_{max}$  denotes the highest profit among all customers. The proposed ratio  $Ratio_{cp}$  encompasses three components that are related respectively to the difference of travel time occurred after the insertion, to the capacity consumption and to the profit collected. For the three components, we associate the specific weights  $\beta$ ,  $\gamma$  and  $\alpha$  to control their relative importance in the ratio. The three components are normalized to be treated in the same manner. Our ratio attempts to maximize the profit and to minimize the resources consumption. Therefore, the insertion having the lowest ratio is selected.

The adaptive construction mechanism consists of generating the best triplet  $(\alpha, \beta, \gamma)$  at each iteration. For this reason, we apply six constructive heuristics  $h_j$ ,  $j \in \{1, \dots, 6\}$  with different combinations of  $(\alpha, \beta, \gamma)$  on six copies of  $S$  by applying BIA. Among the six resulting solutions, the solution having the maximum total profit is retained. Starting from a partial or an empty solution  $S$ , we initialized  $\alpha$ ,  $\beta$  and  $\gamma$  to

0.5 then assigning alternate values of  $[0, 1]$  by slightly modifying the last chosen ones from the previous iteration. In the first five heuristics,  $\alpha$  is set to 1 since the objective of our problem is to maximize the total collected profit and the profit of the chosen customer must be more crucial when taking decisions.

The new combinations of  $\beta$  and  $\gamma$  are chosen among the four vertices of the square having the previous combination as a center and a side length equal to 0.1 (Ben-Said et al. 2016), in a way that all three parameters keep their values in  $[0, 1]$ . The fifth heuristic chooses randomly the values of  $\beta$  and  $\gamma$  in  $[0, 1]$  while keeping alpha set to 1. Finally, the last heuristic chooses randomly all three parameters between 0 and 1. Each heuristic returns the best solution obtained and saves its corresponding triplet  $(\alpha, \beta, \gamma)$  for the next iteration. The aim of the adaptive construction mechanism is to perform separately parallel searches in the neighborhood of the current solution which lead to faster convergence toward solutions with good quality.

In order to escape from local optima, a perturbation phase must be applied on the solution. Therefore, at the end of each construction phase, a random number of customers is removed from the resulting solution. Initially, the number of customers removed,  $d_{max}$ , is limited to 3, then, at each non-improving iteration,  $d_{max}$  is incremented by one until reaching the value of  $n/m$ , which represents the average number of customers per route (Afifi et al. 2016). The aim of the adaptive destruction mechanism is to discover more distant solutions in the search space. Therefore, when improving a certain solution, we reset  $d_{max}$  to its initial value 3 in order to thoroughly explore the neighborhood of the new optimum.

Our algorithm AIDCH starts with a set of empty routes and initializes the parameters of the construction and the destruction steps. At each iteration of AIDCH, the construction phase is applied until reaching a local optimum  $S$ . If  $S$  is the best solution found so far, it is retained as the result of AIDCH. This step is immediately followed by the destruction phase, where a limited random number of customers is removed from the obtained solution. To arrange the solution and reduce the travel time, a 2-opt operator is applied. These steps are repeated until reaching the maximum number of iterations without improvement of the best solution. The stopping condition set to  $n$  and the maximum number of customers to be removed set to  $n/m$  are two predefined parameters for our AIDCH (Ben-Said et al. 2016).

### 3.3 Local search: routes search space

To improve the quality of CTOP solutions in the *routes* search space, several local searches are applied. We performed two versions of our VSS algorithm, the first one is done using the Simulated Annealing as one of the local searches, and the other version is by applying a Tabu Search.

#### 3.3.1 Simulated annealing

The simulated Annealing procedure is involved in the *routes* search space. After extracting a solution from a giant tour, our SA algorithm tries to improve the quality of the new solution. It explores the neighborhood of a solution by applying a fast version



of the AIDCH algorithm. The adaptive destruction and construction mechanisms are used. These operators aim to maximize the total collected profit by removing some customers from the solution in order to insert other customers that increase the profit of the resulting solution. Furthermore, the 2-opt operator is applied to reduce the total travel time by exchanging two arcs belonging to the same route. Each time one neighbor is produced, it will be accepted with a certain probability. AIDCH is applied to the child  $S$ . If the resulting solution  $S'$  improves  $S$ , it replaces the child  $S$  with the best solution  $S_{SA}$  in the next iterations of the local search. Otherwise, SA accepts  $S'$  with the probability  $e^{-\frac{\Delta_p}{Temp}}$ , where  $Temp$  represents the current temperature and  $\Delta_p = P(S) - P(S')$  the difference between the total collected profit in the solution  $S$  and the modified one  $S'$ . For each child  $S$ , the temperature  $Temp$  is initialized with the initial temperature  $Temp_0$  and evolves during the exploration of the neighborhood of  $S$  with a geometric progression. After each iteration, we update  $Temp$  with the value  $k*Temp_0$ , where  $k$  is the cooling speed. In Sect. 4.3 we present the computational techniques used to calculate  $Temp_0$  and to choose the best value of  $k$ . At the end, SA algorithm stops when all neighborhoods fail to produce any new solution to replace the child  $S$ .

### 3.3.2 Tabu search

In the same perspective, a Tabu Search (TS) is embedded in our algorithm in order to avoid to be trapped in local optima. Compared to the basic local search where only moves towards improved solutions are permitted, TS may accept degraded solutions without falling back into recently emerged ones. For this purpose, in our implementation, when a customer  $c$  visited between customers  $i$  and  $j$  is removed from the solution, a temporary tabu status is assigned to the insertion of  $c$  in the position  $(i, j)$ . TS makes use of a short term memory (known as tabu list) to keep a track on the most recently tabu moves and forbid applying them for some upcoming iterations. The tabu status is maintained as in Gendreau et al. (1998) for  $\theta$  iterations, where  $\theta$  is randomly chosen in the range  $[5, 25]$  within a discrete uniform distribution. Since the tabu status may forbid some interesting moves probably leading to a new solution better than the current local optimum, we use an aspiration criterion to accept solutions containing tabu insertions only if they present an evaluation higher than the best solution found so far. The overall procedure iterates for around 10 iterations without getting any further improvement.

### 3.4 Giant tours search space

After providing a starting solution by GRASP, ELS algorithm alternates between the two spaces, the *routes* search space and the *giant tours* search space. In the later one, several local searches are applied on the sequence of customers in order to improve the quality of the extracted solution using our optimal split procedure.

### 3.4.1 Split procedure

Our VSS algorithm performs several local searches in the *giant tour* search space to improve the quality of the sequence. In order to evaluate the resulting sequence, we developed an optimal split procedure adapted from the split procedure proposed by Dang et al. (2013) for TOP. The objective of the split is to extract from a given sequence, the CTOP optimal solution composed of  $m$  routes that respect all the resource limitation constraints.

Let  $T = (T[1], \dots, T[n])$  represents the sequence of  $n$  customers. Our optimal split procedure aims to extract from the sequence  $T$ ,  $m$  subsequences that do not have any common customer in between. Choosing the subsequences must respect at the same time the *C-constraint* and the *T-constraint* and in such a way that the total collected profit from the visited customers is maximized.

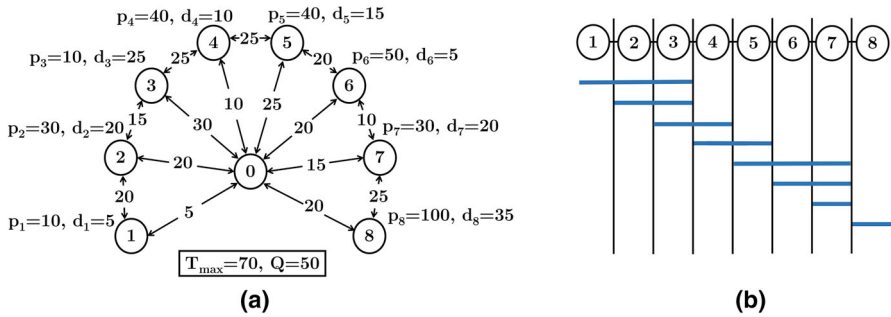
The order of the customers in the chosen subsequences must be followed; therefore, we can identify the subsequences by their starting customers and their lengths. For this reason, we define the subsequence by  $\langle i, \ell_i \rangle_T$  composed of  $\ell_i$  consecutive customers starting with customer  $T[i]$  in the sequence  $T$ . The final solution is then represented by the routes  $\langle (i_1, \ell_{i_1})_T, \dots, (i_k, \ell_{i_k})_T \rangle$ , where  $k \leq m$  and  $i_{p+1} > i_p + \ell_{i_p}$  for all  $p \in \{1, \dots, k-1\}$ .

The first step of our split procedure is to extract all feasible routes starting from all customers in the sequence  $T$ , then choosing among them, the  $m$  routes that maximize the total collected profit. As proven in Dang et al. (2013) for TOP, enumerating only *saturated routes* is relevant to obtain the optimal solution extracted from a given giant tour. Therefore, we define the *saturated route* as the longest feasible route that complies the *C-constraint* and the *T-constraint*.

**Definition 3.1** Starting with the customer  $T[i]$  in the giant tour, the corresponding *saturated route*  $(T[i], \dots, T[i + l_i^{max}])$  is obtained by including all customers following  $T[i]$  in  $T$  as long as the *C - constraint* and the *T - constraint* are respected.

By limiting our extraction to only saturated routes, we start our splitting procedure by extracting the  $n$  saturated routes from the giant tour in order to select the  $m$  routes that do not have any common customers among them. This problem can be formulated as a knapsack problem with conflicts (KPC) with a maximum volume limited to  $m$  (Yamada et al. 2002). The items of KPC represent the  $n$  enumerated routes having each a unitary volume and an item's weight that represents the profit of the route. The existence of common customers between routes leads to a conflict between the corresponding items. Therefore, we modeled our problem using an interval graph, where each saturated route  $(T[i], \dots, T[i + l_i^{max}])$  represents an interval  $[i, i + l_i^{max}]$  using the positions of its customers in the giant tour. The conflict between two routes is represented in the graph with an arc linking these routes. As proposed by Dang et al. (2013), the resulting KPC can be solved using dynamic programming in  $O(m \cdot n)$  time and space.

Figure 2 depicts the mapping of a split problem to a KPC. A giant tour of 8 customers is given in the first graph. In this problem, the maximum travel time is set to 70 and the maximum capacity to 50 for each vehicle among the two available ones. Fig. 2b shows the interval model. Eight saturated routes are extracted. As an example, the first



**Fig. 2** The extraction of saturated subtours from a giant tour. **a** A split problem, **b** the weighted interval model

saturated route  $T_1 = (1, 2, 3)$  has its time  $T(T_1) = 70$  and its capacity  $D(T_1) = 50$ . We cannot include the customer 4 in this route since it violates the  $T - constraint$  and the  $C - constraint$ .

At the end, solving KPC leads to determining among the  $n$  feasible extracted routes, the  $m$  ones that maximize the total profit of the solution, which will represent the result of our split procedure.

### 3.4.2 Local search giant tour space

During the giant tours space exploration, we apply in our ELS algorithm two local search operators to improve the quality of the sequence: the swap and the Construction/Destruction operators. At each iteration of our ELS algorithm, the two operators are applied to the giant tour in a random order. Then, the sequence is evaluated using the optimal split procedure to extract the solution from the giant tour. When at least one of the operators makes an improvement in terms of profit on the extracted solution, the local search procedure reiterates on the new sequence. Otherwise, when both operators fail to improve the quality of the sequence, the local search procedure stops. These two operators work as follows:

- *swap operator* It evaluates each sequence obtained by exchanging the positions of customers in the giant tour. We limited the number of swapped customers to at most 10, chosen randomly from the sequence, in order to reduce the computational time. Each customer is first removed from the sequence, then inserted in all other positions in the sequence while evaluating each resulting sequence by extracting the optimal solution from it. If the resulting solution is improved, the sequence is reinitialized with the new solution to reiterate the local search phase, otherwise a new couple of customers is considered for the swap.
- *construction/destruction operator* First, CTOP solution is extracted from the giant tour using our optimal split procedure. This operator is applied on CTOP solution by removing some customers from the routes then trying to insert the unrouted customers in the solution. The two adaptive mechanisms of destruction and construction described in Sect. 3.2.1 are used.

The extracted solution derived from each modified sequence is then improved using the local searches of the *routes* search space as described in Sect. 3.3.

## 4 Computational experiments

Our VSS Algorithm is coded in C++ and compiled in a Linux environment using GNU GCC. Computational experiments are carried out on an Intel Xeon 2.6 GHz processor. In the following sections, we show some experiments performed to tune the parameters of our algorithm, then we examine the effect of each component, to end up by giving a detailed comparison to the literature on all instances of the CTOP benchmark.

### 4.1 Benchmark of CTOP

We tested our algorithm on the benchmark of CTOP composed of two sets of instances.

- The first dataset was proposed by Archetti et al. (2009) and was deduced from ten instances of the benchmark of the Capacitated VRP (Christofides et al. 1979). The number of customers in each instance ranges between 50 and 99. For each customer  $c$ , its profit  $p_c$  is set to  $(0.5 + h) * d_c$ , where  $d_c$  is the customer's demand and  $h$  is a random number selected uniformly in the interval  $[0, 1]$ . By considering different combinations of the maximum travel time  $T_{max}$ , the vehicle capacity  $Q$  and the fleet size  $m$ , 13 subsets of 10 CTOP instances each are generated.
- The second dataset was introduced by Tarantilis et al. (2013). This set was deduced from ten instances from the benchmark of the Periodic VRP (Pirkwieser and Raidl 2010) using the same technique proposed by Archetti et al. (2009). In this dataset, all instances are large, where the number of customers ranges from 337 to 577. In total, a new large scale benchmark of 130 instances are proposed for CTOP.

The characteristics of the instances are shown in Table 1, where we present for each subset of instances, the number of available vehicles  $m$ , the capacity limit  $Q$  and the maximum travel time  $T_{max}$  for each vehicle.

For convenience, we represent each instance with a label "*base* -  $n$  -  $m$  -  $Q$  -  $T_{max}$ ", where *base* is the name of the original instance taken from the VRP benchmark,  $n$  the number of customers,  $m$  the number of vehicles,  $Q$  the capacity of the vehicles and  $T_{max}$  the maximum travel time of each vehicle.

### 4.2 Protocol and performance metrics

Since VSS is a randomized search method, we applied the same protocol used in Dang et al. (2013) and Zhang et al. (2013), where we solved each instance ten times and we retained the best result obtained among the ten executions, denoted by  $Z_{max}$ .

To evaluate all methods performed for CTOP, we denote the best result obtained in the literature by  $Z_{best}$ , and we define the relative percentage error (RPE) as follows: 
$$RPE = 100 * \frac{Z_{best} - Z_{max}}{Z_{best}}.$$

**Table 1** Benchmark of CTOP

Set	Instances (Archetti et al. 2009)			Instances (Tarantilis et al. 2013)		
	m	Q	$T_{max}$	m	Q	$T_{max}$
1	10–20	140–200	160–1040	14–24	330–375	660–720
2	2	50	50	6	75	100
3	3	50	50	7	75	100
4	4	50	50	8	75	100
5	2	75	75	6	150	200
6	3	75	75	7	150	200
7	4	75	75	8	150	200
8	2	100	100	6	200	400
9	3	100	100	7	200	400
10	4	100	100	8	200	400
11	2	140–200	160–1040	6	330–375	660–720
12	3	140–201	160–1041	7	330–376	660–721
13	4	140–202	160–1042	8	330–377	660–722

The performance time of the methods is evaluated using the Time To Best (TTB) protocol used in Archetti et al. (2009), Tarantilis et al. (2013), and Luo et al. (2013). TTB is defined as the CPU time required to reach the best solution within one run.

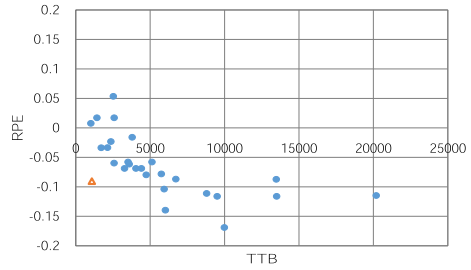
### 4.3 Parameter tuning

In our proposed method there are three main parameters that must be well chosen in order to get the best performance of our VSS framework. Two other parameters are to be set in the SA component. We carried out the parameter tuning experiments using a subset of 5 instances chosen randomly among the large instances of the benchmark of CTOP. The instances are “4-481-8-75-100”, “1-337-6-150-200”, “3-433-6-150-200”, “8-433-7-200-400” and “2-385-6-350-713”.

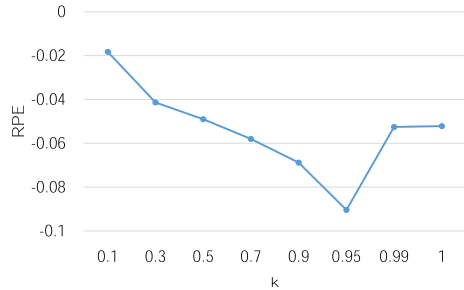
#### 4.3.1 VSS parameters

Starting with our VSS framework, three parameters should be well tuned to obtain the best results for our overall algorithm. These parameters are the number of initial solutions  $n_{init}$  given to GRASP, the number of child solutions  $n_{child}$  and the number of iterations  $n_{els}$  of the evolutionary local search. Several tests have been launched to determine the values of these parameters offering the best compromise between the quality of the obtained solutions and the computational time. Therefore we build a Pareto test with different combinations of  $n_{init}$ ,  $n_{child}$  and  $n_{els}$  ranged from (5, 5, 5) up to (15, 15, 15) conducting to a total number of 27 tests. In these tests, the two other parameters  $Temp_0$  and  $k$  were set based on the studies of Afifi et al. (2016). The results of parameters combinations are depicted in Fig. 3, where we plot for each

**Fig. 3** Pareto front of results obtained with different parameters combinations



**Fig. 4** Effect of the  $k$  parameter in the VSS-SA performance



combination the average RPE and the average TTB of the solutions obtained for the 5 tested instances.

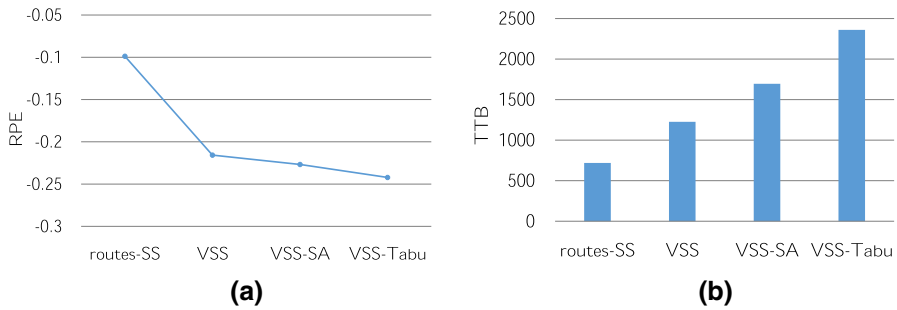
Based on the results obtained in Fig. 3, we have chosen the configuration that gives the best compromise between the quality of the solutions and the computational time. This configuration is represented by the point covering the largest volume in the objective space. The resulting parameters are:  $n_{init} = 5$ ,  $n_{child} = 10$  and  $n_{els} = 10$ . We note that, for the rest of our tests, we used this configuration in our VSS framework.

### 4.3.2 Simulated annealing parameters

The objective of applying SA algorithm in our approach is to accept non-improved solutions at the beginning of the solution process before starting to completely reject them after several iterations. For this reason, the initial temperature  $Temp_0$  is calculated in such a way to obtain a high probability of accepting a non-improved solution at the beginning of each GRASP iteration is sufficiently high. This probability is set to 0.95. During AIDCH, we record the maximum and the minimum profit values obtained denoted by  $score_{max}$  and  $score_{min}$  respectively.  $Temp_0$  is then computed from the following expression:  $e^{-\frac{\Delta_{p0}}{Temp_0}} = 0.95$  (Afifi et al. 2016), where  $\Delta_{p0} = score_{max} - score_{min}$ .

The second parameter that should be tuned in the SA algorithm is the cooling speed  $k$ . In order to choose the best value of  $k$ , we performed several tests with different values of  $k$  chosen from the subset  $\{0.1, 0.3, 0.5, 0.7, 0.9, 0.95, 0.99, 1\}$ . Figure 4 represents the evolution of the average RPE with respect to value of  $k$ .

The convex shape of the graph shown in Fig. 4 explains the fact that choosing a small or a substantially high value of  $k$  leads to a decrease in the performance of



**Fig. 5** Performance of the combinations of the components of the VSS. **a** Analysis of the average RPE, **b** analysis of the time TTB

VSS-SA algorithm. This behavior is due to the fact that when choosing a small value for  $k$ , our SA algorithm might stop accepting non-improved solutions in the early stage, leading to local optimum. Moreover, choosing a big value for  $k$  increases the probability of accepting non-improved solutions all the time, which might cause the intense diversification of the solutions. Therefore, we have chosen  $k = 0.95$  since it gave us the best average RPE.

#### 4.4 Components evaluation

In this section, we evaluate the effectiveness of each component of our algorithm. First we tested our algorithm using only the *routes* search space, then we considered the basic VSS framework, with several other alternative versions where in each version one additional component is evaluated as follows.

**routes-SS** it represents our global scheme in which only CTOP *routes* search space is explored.

**VSS** this is the basic version. VSS scheme is launched while considering CTOP *routes* search space and *giant tour* search space.

**VSS-SA** this version is the combination of VSS with SA component.

**VSS-Tabu** it represents the combination of VSS with the tabu component.

We tested these versions with the same parameters configuration. Figure 5 summarizes the performance of each method, where we compared their average RPE and TTB.

Based on the results obtained in Fig. 5, we notice that the average RPE obtained with the version VSS is much better than that obtained with *routes-SS* which leads us to deduce that exploring the two search spaces provides a better performance for our VSS algorithm. Finally, incorporating SA and the tabu components into our VSS algorithm hugely improves the quality of the obtained solutions, although the two alternatives consume slightly more computational time.

#### 4.4.1 Comparison with the literature on the first dataset (Archetti et al. 2009)

The exact method proposed by Archetti et al. (2009) was able to find the optimal solutions for some of the small instances of the first dataset. These results were later outperformed by the branch and price algorithm proposed by Archetti et al. (2013). For the remaining instances that do not have an optimal solution yet, we compare our results to those obtained by the best heuristic methods proposed in the literature, which are the VNS algorithm proposed by Archetti et al. (2009), the BiF&F algorithm proposed by Tarantilis et al. (2013) and the ADEPT-RD algorithm of Luo et al. (2013). We present in Table 2 the detailed results obtained by these three methods and our two algorithms. In this table, column  $Z_{best}$  represents the best known results obtained in the literature, where the values with an asterisk are the optimal solutions. For each method, the column  $Z_{max}$  provides the profit of the best solution given for each instance, column RPE reports the relative percentage error from the best known solutions and column TTB represents the time to best which is the time (in seconds) retained when the best solution is obtained. We used dash mark ‘–’ when the corresponding method did not provide a solution for the instance. The improvements performed by our algorithms VSS-Tabu and VSS-SA are marked in bold. Finally, the last line of the table presents the average TTB and the average RPE for all methods. To compare the cpu performance of the different machines used to test these methods, we used the recommended Super PI protocol (Luo et al. 2013). Super PI evaluates the computational power by running in all machines a single threaded program to compute a specific number of digits of  $\pi$ . We note that VNS algorithm of Archetti et al. was run on a personal computer Intel Pentium 4 CPU 2.80 GHz and 1.048 GB Ram. BiF&F of Tarantilis et al. was performed on a 2.83 Giga Hertz Intel Core2 Quad PC over a single thread, while Luo et al. mentioned in their paper that their CPU is roughly four times faster than that of Archetti et al. according to the publicly available SuperPi benchmark. Our Computational experiments are carried out on an Intel Xeon 2.6 GHz processor. According to the obtained scores on Super PI, we notice that our machine presents almost similar cpu speed as the machines used by Tarantilis et al. (2013) and Luo et al. (2013) and outperforms by three times the machine used by Archetti et al. (2009).

Based on the results obtained in Table 2 the competitiveness of our methods is clearly noticed with respect to the other best heuristic methods available in the literature. Our two algorithms VSS-Tabu and VSS-SA provide all the best known results in the literature with the smallest TTB. Furthermore, they improve the solution quality of one of the hardest instances of the benchmark, where they both reach a relative percentage error of  $-0.0003\%$  with respect to the best known result in the literature. More precisely, this table reveals that our algorithms achieve better results compared to those found by the VNS algorithm which registered an average RPE of  $0.08\%$ . Even with the difference in cpu speed between our machine and the one used by Archetti et al. (2009), the TTB of our algorithms is substantially lower than that of the VNS algorithm. Moreover, our results exceed those scored by BiF&F algorithm which resulted in an average RPE of  $0.01\%$ . In terms of computational time performance, the TTB of VSS-Tabu is slightly higher than that of BiF&F algorithm but the TTB of VSS-SA is much lower. Furthermore, our algorithms obtain the same solutions obtained by ADEPT-RD algorithm—the best known results of the literature. We



**Table 2** Comparison between our results and those obtained in the literature on the first dataset (Archetti et al. 2009)

Instance	$Z_{best}$			BiF&F			ADEPT-RD			VSS-Tabu			VSS-SA		
	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
03-101-15-200-200	1409*	0	0	1409	0	0	-	-	-	1409	0	0	1409	0	2
06-51-10-160-200	761*	0	0	761	0	0	-	-	-	761	0	0	761	0	0
07-76-20-140-160	1327*	0	0	1327	0	0	-	-	-	1327	0	0	1327	0	1
08-101-15-200-230	1409*	0	0	1409	0	0	-	-	-	1409	0	0	1409	0	1
09-151-10-200-200	2065	0.05	3600	2065	0	2	-	-	-	2065	0	39	2065	0	120
10-200-20-200-200	3048*	0	0	3048	0	0	-	-	-	3048	0	0	3048	0	11
13-121-15-200-720	1287*	0	0	1287	0	0	-	-	-	1287	0	0	1287	0	2
14-101-10-200-1040	1710*	0	0	1710	0	0	-	-	-	1710	0	0	1710	0	3
15-151-15-200-200	2159*	0	0	2159	0	0	-	-	-	2159	0	0	2159	0	7
16-200-15-200-200	2968	0	3600	-	-	-	-	-	-	<b>2969</b>	-0.0337	61	<b>2969</b>	-0.0337	254
03-101-2-50-50	133*	0	46	133	0	0	133	0	2	133	0	0	133	0	0
06-51-2-50-50	121*	0	5	121	0	0	121	0	1	121	0	0	121	0	0
07-76-2-50-50	126*	0	17	126	0	0	126	0	3	126	0	0	126	0	0
08-101-2-50-50	133*	0	46	133	0	0	133	0	2	133	0	0	133	0	0
09-151-2-50-50	137*	0	192	137	0	0	137	0	3	137	0	0	137	0	0
10-200-2-50-50	134*	0	423	134	0	0	134	0	11	134	0	0	134	0	0
13-121-2-50-50	134*	0	21	134	0	0	134	0	0	134	0	0	134	0	0
14-101-2-50-50	124*	0	46	123	0.81	0	124	0	2	124	0	0	124	0	0
15-151-2-50-50	134*	0	166	134	0	2	134	0	8	134	0	6	134	0	3
16-200-2-50-50	137*	0	422	137	0	0	137	0	3	137	0	0	137	0	0
03-101-3-50-50	198*	0	63	198	0	0	198	0	38	198	0	0	198	0	0
06-51-3-50-50	177*	0	8	177	0	0	177	0	1	177	0	0	177	0	0

Table 2 continued

Instance	$Z_{best}$	VNS			BiF&F			ADEPT-RD			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
07-76-3-50-50	187*	187	0	27	187	0	0	187	0	0	187	0	0	187	0	0
08-101-3-50-50	198*	198	0	59	198	0	0	198	0	38	198	0	0	198	0	0
09-151-3-50-50	201*	201	0	227	201	0	0	201	0	3	201	0	0	201	0	0
10-200-3-50-50	200*	200	0	549	200	0	0	200	0	2	200	0	0	200	0	1
13-121-3-50-50	193*	193	0	30	193	0	0	193	0	0	193	0	0	193	0	0
14-101-3-50-50	184*	184	0	67	184	0	0	184	0	3	184	0	0	184	0	0
15-151-3-50-50	200*	200	0	214	200	0	2	200	0	5	200	0	0	200	0	1
16-200-3-50-50	203*	203	0	580	203	0	0	203	0	4	203	0	0	203	0	0
03-101-4-50-50	260*	260	0	89	260	0	0	260	0	1	260	0	0	260	0	0
06-51-4-50-50	222*	222	0	13	222	0	0	222	0	0	222	0	0	222	0	0
07-76-4-50-50	240*	240	0	40	240	0	0	240	0	1	240	0	0	240	0	0
08-101-4-50-50	260*	260	0	88	260	0	0	260	0	1	260	0	0	260	0	0
09-151-4-50-50	262*	262	0	293	262	0	4	262	0	45	262	0	0	262	0	1
10-200-4-50-50	265*	265	0	788	265	0	4	265	0	37	265	0	0	265	0	1
13-121-4-50-50	243*	243	0	47	243	0	0	243	0	1	243	0	0	243	0	1
14-101-4-50-50	241*	241	0	97	241	0	0	241	0	2	241	0	0	241	0	0
15-151-4-50-50	266*	266	0	305	266	0	1	266	0	19	266	0	0	266	0	2
16-200-4-50-50	269*	269	0	677	269	0	1	269	0	9	269	0	0	269	0	0
03-101-2-75-75	208*	208	0	383	208	0	0	208	0	0	208	0	0	208	0	0
06-51-2-75-75	183*	183	0	53	183	0	2	183	0	55	183	0	1	183	0	1
07-76-2-75-75	193*	193	0	149	193	0	0	193	0	1	193	0	0	193	0	0

Table 2 continued

Instance	$Z_{best}$	VNS			BiF&F			ADEPT-RD			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
08-101-2-75-75	208*	208	0	383	208	0	0	208	0	0	208	0	0	208	0	0
09-151-2-75-75	210*	210	0	1166	210	0	1	210	0	20	210	0	0	210	0	0
10-200-2-75-75	208*	208	0	2596	208	0	5	208	0	8	208	0	0	208	0	0
13-121-2-75-75	193*	193	0	11	193	0	0	193	0	0	193	0	0	193	0	0
14-101-2-75-75	190*	190	0	140	190	0	0	190	0	1	190	0	0	190	0	0
15-151-2-75-75	211*	210	0.47	1240	211	0	48	211	0	23	211	0	6	211	0	14
16-200-2-75-75	212*	212	0	2808	212	0	0	212	0	24	212	0	0	212	0	0
03-101-3-75-75	307*	307	0	500	307	0	0	307	0	1	307	0	0	307	0	0
06-51-3-75-75	269*	269	0	71	269	0	0	269	0	4	269	0	0	269	0	0
07-76-3-75-75	287*	287	0	183	287	0	1	287	0	1	287	0	0	287	0	1
08-101-3-75-75	307*	307	0	500	307	0	0	307	0	1	307	0	0	307	0	0
09-151-3-75-75	312*	310	0.64	1724	312	0	19	312	0	9	312	0	17	312	0	17
10-200-3-75-75	311*	310	0.32	3122	311	0	36	311	0	38	311	0	1	311	0	15
13-121-3-75-75	265*	265	0	18	265	0	0	265	0	0	265	0	0	265	0	0
14-101-3-75-75	279*	279	0	201	279	0	0	279	0	0	279	0	0	279	0	0
15-151-3-75-75	315*	315	0	1593	315	0	1	315	0	143	315	0	0	315	0	0
16-200-3-75-75	317*	317	0	3600	317	0	2	317	0	134	317	0	0	317	0	1
03-101-4-75-75	403*	401	0.50	685	403	0	29	403	0	67	403	0	32	403	0	11
06-51-4-75-75	349*	349	0	96	349	0	0	349	0	14	349	0	0	349	0	1
07-76-4-75-75	378*	378	0	265	378	0	17	378	0	8	378	0	0	378	0	2
08-101-4-75-75	403*	401	0.50	685	403	0	29	403	0	68	403	0	10	403	0	11
09-151-4-75-75	408*	407	0.25	1862	408	0	60	408	0	6	408	0	39	408	0	24
10-200-4-75-75	411*	410	0.24	3600	411	0	135	411	0	88	411	0	70	411	0	12

Table 2 continued

Instance	$Z_{best}$	VNS			BIF&F			ADEPT-RD			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
13-121-4-75-75	323*	323	0	23	323	0	0	323	0	0	323	0	0	323	0	0
14-101-4-75-75	366*	366	0	316	366	0	3	366	0	0	366	0	0	366	0	2
15-151-4-75-75	415*	414	0.24	2118	415	0	20	415	0	67	415	0	2	415	0	10
16-200-4-75-75	420*	419	0.24	3600	420	0	0	420	0	76	420	0	4	420	0	15
03-101-2-100-100	277*	277	0	472	277	0	10	277	0	27	277	0	4	277	0	4
06-51-2-100-100	252*	252	0	60	252	0	0	252	0	20	252	0	0	252	0	0
07-76-2-100-100	266*	266	0	163	266	0	0	266	0	2	266	0	0	266	0	0
08-101-2-100-100	277*	277	0	472	277	0	10	277	0	28	277	0	5	277	0	3
09-151-2-100-100	279*	279	0	1646	279	0	4	279	0	66	279	0	0	279	0	0
10-200-2-100-100	282*	282	0	3111	282	0	53	282	0	5	282	0	0	282	0	4
13-121-2-100-100	253	253	0	76	253	0	0	253	0	2	253	0	0	253	0	0
14-101-2-100-100	271*	271	0	276	271	0	0	271	0	32	271	0	0	271	0	2
15-151-2-100-100	282*	282	0	1672	282	0	4	282	0	7	282	0	0	282	0	0
16-200-2-100-100	285*	284	0.35	3523	285	0	56	285	0	168	285	0	9	285	0	41
03-101-3-100-100	408*	407	0.25	685	407	0.25	11	408	0	423	408	0	20	408	0	9
06-51-3-100-100	369*	369	0	92	369	0	0	369	0	2	369	0	0	369	0	0
07-76-3-100-100	397*	397	0	233	397	0	19	397	0	22	397	0	0	397	0	3
08-101-3-100-100	408*	407	0.25	685	407	0.25	11	408	0	429	408	0	18	408	0	21
09-151-3-100-100	415*	413	0.48	2151	415	0	96	415	0	0	415	0	6	415	0	41
10-200-3-100-100	418	416	0.48	3600	418	0	169	418	0	15	418	0	123	418	0	25
13-121-3-100-100	344	344	0	113	344	0	0	344	0	0	344	0	0	344	0	1
14-101-3-100-100	399*	399	0	406	399	0	0	399	0	302	399	0	3	399	0	8
15-151-3-100-100	418*	417	0.24	1966	418	0	36	418	0	12	418	0	58	418	0	45

Table 2 continued

Instance	$Z_{best}$	VNS			BiF&F			ADEPT-RD			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
16-200-3-100-100	423*	420	0.71	3600	423	0	201	423	0	54	423	0	55	423	0	52
03-101-4-100-100	532*	529	0.56	963	532	0	42	532	0	313	532	0	27	532	0	12
06-51-4-100-100	482*	481	0.21	135	482	0	0	482	0	2	482	0	0	482	0	2
07-76-4-100-100	521*	521	0	342	521	0	16	521	0	17	521	0	1	521	0	3
08-101-4-100-100	532*	529	0.56	963	532	0	41	532	0	314	532	0	29	532	0	20
09-151-4-100-100	546*	545	0.18	2934	546	0	38	546	0	47	546	0	53	546	0	31
10-200-4-100-100	553*	548	0.90	3600	553	0	243	553	0	216	553	0	11	553	0	40
13-121-4-100-100	419	419	0	179	419	0	0	419	0	0	419	0	0	419	0	1
14-101-4-100-100	525*	525	0	670	525	0	0	525	0	5	525	0	0	525	0	1
15-151-4-100-100	549*	548	0.18	2828	549	0	206	549	0	66	549	0	25	549	0	49
16-200-4-100-100	558*	554	0.72	3600	558	0	67	558	0	5	558	0	115	558	0	79
03-101-2-200-200	536*	536	0	305	536	0	1	536	0	0	536	0	10	536	0	5
06-51-2-160-200	403*	403	0	27	403	0	0	403	0	2	403	0	0	403	0	0
07-76-2-140-160	377*	377	0	107	377	0	0	377	0	6	377	0	26	377	0	1
08-101-2-200-230	536*	536	0	211	536	0	1	536	0	0	536	0	21	536	0	4
09-151-2-200-200	548	547	0.18	952	548	0	5	548	0	0	548	0	107	548	0	32
10-200-2-200-200	556	556	0	2214	556	0	5	556	0	0	556	0	158	556	0	5
13-121-2-200-720	513	513	0	253	513	0	0	513	0	4	513	0	1	513	0	1
14-101-2-200-1040	534*	534	0	109	534	0	0	534	0	0	534	0	0	534	0	0
15-151-2-200-200	550	550	0	929	550	0	0	550	0	8	550	0	1	550	0	1
16-200-2-200-200	558	558	0	2040	558	0	0	558	0	0	558	0	1	558	0	2
03-101-3-200-200	762	762	0	643	762	0	14	762	0	13	762	0	3	762	0	2
06-51-3-160-200	565*	565	0	43	565	0	0	565	0	0	565	0	0	565	0	0

Table 2 continued

Instance	$Z_{best}$	VNS			BIF&F			ADEPT-RD			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
07-76-3-140-160	548*	548	0	164	548	0	8	548	0	1	548	0	7	548	0	14
08-101-3-200-230	762	762	0	482	762	0	0	762	0	6	762	0	1	762	0	1
09-151-3-200-200	797	796	0.13	1805	797	0	1	797	0	18	797	0	90	797	0	41
10-200-3-200-200	816	815	0.12	3467	816	0	1	816	0	7	816	0	14	816	0	29
13-121-3-200-720	727	727	0	557	727	0	0	727	0	14	727	0	7	727	0	6
14-101-3-200-1040	770*	770	0	210	770	0	0	770	0	0	770	0	0	770	0	1
15-151-3-200-200	802	801	0.12	1711	802	0	56	802	0	0	802	0	306	802	0	136
16-200-3-200-200	822	821	0.12	3360	822	0	0	822	0	43	822	0	324	822	0	81
03-101-4-200-200	950	950	0	961	950	0	0	950	0	12	950	0	16	950	0	11
06-51-4-160-200	683*	683	0	53	683	0	0	683	0	10	683	0	0	683	0	1
07-76-4-140-160	707*	707	0	296	707	0	5	707	0	0	707	0	2	707	0	1
08-101-4-200-230	950	950	0	726	950	0	1	950	0	10	950	0	26	950	0	8
09-151-4-200-200	1033	1033	0	2903	1033	0	2	1033	0	7	1033	0	44	1033	0	11
10-200-4-200-200	1064	1064	0	3600	1064	0	1	1064	0	0	1064	0	21	1064	0	15
13-121-4-200-720	908	908	0	954	908	0	0	908	0	1	908	0	217	908	0	27
14-101-4-200-1040	975*	975	0	483	975	0	1	975	0	0	975	0	0	975	0	1
15-151-4-200-200	1031	1031	0	2832	1031	0	3	1031	0	112	1031	0	262	1031	0	58
16-200-4-200-200	1073	1073	0	3600	1073	0	1	1073	0	2	1073	0	40	1073	0	15
Average			0.08	952		0.01	14		0	33		-0.0003	20		-0.0003	12

found that VSS-Tabu algorithm outperforms ADEPT-RD in terms of computational time while VSS-SA outperforms it within a TTB almost 3 times smaller than the one consumed by ADEPT-RD.

#### 4.4.2 Comparison with the literature on the second dataset (Tarantilis et al. 2013)

Tarantilis et al. (2013) proposed two heuristic methods, which are the slow and the fast versions of the Bi-Level Filter and Fan method BiF&F and tested their algorithms on a new large scale instances. Table 3 summarizes the comparison between our VSS-Tabu and VSS-SA algorithms with the fast and the slow versions of the BiF&F method. In this table column *Instance* represents the name of the instance, and column  $Z_{best}$  presents the best known solution in the literature. To compare our results to those of BiF&F-f and BiF&F-s algorithms, this table presents the maximum score obtained by each method  $Z_{max}$  and the CPU time in seconds elapsed to identify the best solution (TTB). We mark in bold all the improvements obtained by our algorithms compared to two versions of BiF&F algorithm.

The results obtained in Table 3 clearly prove the competitiveness of our VSS-Tabu and VSS-SA algorithms in terms of quality of the obtained solutions and computational time consumed during the solution process. We notice from Table 3 that our VSS-Tabu algorithm reached the best known results for 119 instances and improved 74 instances among the 130 of the second dataset. Compared to the fast version of BiF&F algorithm, VSS-Tabu presents 91 strict improvements. To achieve these performances VSS-Tabu requires a higher TTB than that of BiF&F-f. Compared to the slow version of BiF&F algorithm, our VSS-Tabu algorithm is able to outperform this method in terms of quality of the solutions and computational time, where the average RPE of our VSS-Tabu was  $-0.242\%$ , compared to the slow BiF&F algorithm that found  $0.02\%$ . To obtain these results, our VSS-Tabu algorithm consumed a lower TTB compared to that consumed by the slow BiF&F algorithm.

Moreover, our VSS-SA algorithm reached a better average RPE of  $-0.227\%$  with respect to the best known results in the literature. More precisely, it obtained the best results for 124 instances and improved the results of 77 solutions. Apart from effectiveness, VSS-SA demonstrates its scalability and speed since it is two times faster compared to BiF&F-s.

Table 4 presents the improvements of each method compared to the other methods separately. This table shows that our VSS-SA and VSS-Tabu algorithms improved the results of the slow BiF&F by respectively 79 and 76 instances and that of the fast BiF&F by 96 and 91 instances, while BiF&F-s algorithm had scored 6 and 11 improvements with respect to our results and BiF&F-f algorithm had found only 3 and 4 improvements.

Overall, the obtained results prove the efficiency and rapidity of our two algorithms compared to all other heuristics proposed to solve CTOP. We can also conclude that they present highly competitive performances. More precisely, VSS-Tabu obtains 27 improvements compared to the results of VSS-SA against 30 improvements for the latter. On the other hand, VSS-Tabu outperforms VSS-SA in term of average RPE since it improves the best known results by  $-0.242\%$  while VSS-SA obtains an average RPE of  $-0.227\%$ .

**Table 3** Comparison between our results and the literature on the second dataset (Tarrantilis et al. 2013)

Instance	$Z_{best}$	BiF&F-f			BiF&F-s			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
01-337-14-345-720	4172	4172	0	17	4172	0	17	4172	0	1	4172	0	2
02-385-16-350-713	4784	4784	0	25	4784	0	25	4784	0	1	4784	0	3
03-433-18-330-675	5201	5201	0	33	5201	0	32	5201	0	2	5201	0	3
04-481-20-335-713	5828	5828	0	48	5828	0	47	5828	0	1	5828	0	3
05-529-22-340-705	6445	6445	0	101	6445	0	98	6445	0	3	6445	0	5
06-577-24-365-683	7071	7071	0	94	7071	0	93	7071	0	1	7071	0	2
07-361-15-335-668	4355	4355	0	24	4355	0	23	4355	0	1	4355	0	3
08-433-18-350-675	5194	5194	0	51	5194	0	49	5194	0	2	5194	0	4
09-505-21-360-660	6183	6183	0	103	6183	0	104	6183	0	3	6183	0	22
10-577-24-375-675	7239	7239	0	144	7239	0	144	7239	0	4	7239	0	7
01-337-6-75-100	460	459	0.22	10	460	0	30	460	0	16	460	0	17
02-385-6-75-100	543	543	0	22	543	0	35	<b>547</b>	-0.74	29	<b>547</b>	-0.737	44
03-433-6-75-100	599	598	0.17	148	599	0	127	599	0	313	599	0	115
04-481-6-75-100	528	524	0.76	2	528	0	78	<b>535</b>	-1.33	55	<b>535</b>	-1.326	32
05-529-6-75-100	493	493	0	11	493	0	11	493	0	54	493	0	15
06-577-6-75-100	592	588	0.68	215	592	0	494	592	0	31	592	0	132
07-361-6-75-100	546	546	0	8	546	0	8	546	0	3	546	0	3
08-433-6-75-100	527	527	0	16	527	0	25	<b>530</b>	-0.57	42	<b>529</b>	-0.380	30
09-505-6-75-100	530	528	0.38	15	530	0	30	<b>531</b>	-0.19	19	<b>531</b>	-0.189	23
10-577-6-75-100	570	570	0	4	570	0	4	<b>574</b>	-0.70	3	<b>574</b>	-0.702	6
01-337-7-75-100	513	512	0.19	16	513	0	85	513	0	6	513	0	8
02-385-7-75-100	624	624	0	27	623	0.16	171	<b>626</b>	-0.32	99	<b>626</b>	-0.321	47



Table 3 continued

Instance	$Z_{best}$	BiF&F-f			BiF&F-s			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
03-433-7-75-100	689	689	0	76	689	0	337	<b>690</b>	-0.15	159	<b>690</b>	-0.145	133
04-481-7-75-100	599	598	0.17	79	599	0	93	<b>613</b>	-2.34	70	<b>613</b>	-2.337	64
05-529-7-75-100	560	560	0	19	560	0	19	560	0	66	560	0	32
06-577-7-75-100	680	680	0	402	680	0	1959	680	0	324	680	0	186
07-361-7-75-100	620	618	0.32	21	620	0	68	620	0	29	620	0	28
08-433-7-75-100	599	599	0	13	599	0	13	<b>605</b>	-1.00	56	<b>604</b>	-0.835	31
09-505-7-75-100	603	603	0	40	603	0	144	603	0	53	603	0	31
10-577-7-75-100	650	650	0	9	650	0	9	<b>652</b>	-0.31	82	<b>652</b>	-0.308	52
01-337-8-75-100	566	557	1.59	20	566	0	95	566	0	28	566	0	26
02-385-8-75-100	694	694	0	7	694	0	7	<b>701</b>	-1.01	79	<b>699</b>	-0.720	37
03-433-8-75-100	777	775	0.26	1	777	0	1021	<b>780</b>	-0.39	133	<b>780</b>	-0.386	114
04-481-8-75-100	685	673	1.75	19	685	0	308	<b>687</b>	-0.29	198	<b>687</b>	-0.292	488
05-529-8-75-100	623	623	0	227	622	0.16	303	623	0	99	623	0	30
06-577-8-75-100	767	765	0.26	106	767	0	257	<b>768</b>	-0.13	362	767	0	223
07-361-8-75-100	689	683	0.87	12	689	0	99	689	0	47	689	0	22
08-433-8-75-100	676	670	0.89	34	676	0	91	<b>678</b>	-0.30	56	<b>678</b>	-0.296	48
09-505-8-75-100	671	671	0	14	671	0	14	<b>672</b>	-0.15	73	<b>672</b>	-0.149	31
10-577-8-75-100	727	721	0.83	26	727	0	50	<b>729</b>	-0.28	78	<b>729</b>	-0.275	38
01-337-6-150-200	1106	1106	0	355	1104	0.18	1144	<b>1107</b>	-0.09	600	<b>1107</b>	-0.090	225
02-385-6-150-200	1129	1125	0.35	1684	1129	0	3635	<b>1137</b>	-0.71	793	<b>1135</b>	-0.531	634
03-433-6-150-200	1201	1197	0.33	912	1201	0	5868	1201	0	1242	1201	0	895
04-481-6-150-200	1186	1186	0	22	1186	0	24	<b>1191</b>	-0.42	1567	<b>1191</b>	-0.422	620

**Table 3** continued

Instance	$Z_{best}$	BiF&F-f			BiF&F-s			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
05-529-6-150-200	1149	1149	0	466	1149	0	2391	1171	-1.91	1595	1169	-1.741	793
06-577-6-150-200	1230	1230	0	525	1230	0	4035	1233	-0.24	1991	1233	-0.244	699
07-361-6-150-200	1141	1139	0.18	836	1141	0	486	1144	-0.26	809	1143	-0.175	268
08-433-6-150-200	1128	1128	0	644	1122	0.53	1131	1131	-0.27	1130	1131	-0.266	503
09-505-6-150-200	1102	1100	0.18	29	1102	0	1221	1106	-0.36	1318	1105	-0.272	843
10-577-6-150-200	1191	1189	0.17	896	1191	0	4159	1198	-0.59	1456	1196	-0.420	930
01-337-7-150-200	1221	1220	0.08	473	1221	0	456	1267	-3.77	586	1262	-3.358	316
02-385-7-150-200	1290	1273	1.32	329	1290	0	5927	1299	-0.70	1100	1298	-0.620	370
03-433-7-150-200	1377	1375	0.15	2192	1377	0	4301	1383	-0.44	2034	1382	-0.363	503
04-481-7-150-200	1357	1357	0	506	1356	0.07	876	1365	-0.59	1527	1365	-0.590	697
05-529-7-150-200	1344	1333	0.82	793	1344	0	10231	1348	-0.30	1696	1346	-0.149	581
06-577-7-150-200	1415	1415	0	5602	1415	0	8870	1421	-0.42	2415	1420	-0.353	1119
07-361-7-150-200	1310	1277	2.52	55	1310	0	4280	1312	-0.15	625	1312	-0.153	323
08-433-7-150-200	1286	1284	0.16	3147	1286	0	2201	1295	-0.70	937	1294	-0.622	585
09-505-7-150-200	1263	1253	0.79	1853	1263	0	4602	1267	-0.32	973	1265	-0.158	653
10-577-7-150-200	1376	1376	0	5514	1372	0.29	1221	1377	-0.07	2355	1376	0	774
01-337-8-150-200	1389	1386	0.22	803	1389	0	1524	1404	-1.08	830	1402	-0.936	255
02-385-8-150-200	1440	1436	0.28	563	1440	0	3271	1454	-0.97	1073	1452	-0.833	456
03-433-8-150-200	1552	1551	0.06	2622	1552	0	2752	1556	-0.26	1672	1557	-0.322	579
04-481-8-150-200	1527	1524	0.20	294	1527	0	9275	1530	-0.20	1455	1531	-0.262	662
05-529-8-150-200	1496	1496	0	1063	1494	0.13	4557	1514	-1.20	1363	1514	-1.203	691
06-577-8-150-200	1602	1588	0.87	1277	1602	0	14037	1607	-0.31	2963	1604	-0.125	985
07-361-8-150-200	1453	1450	0.21	296	1453	0	626	1465	-0.83	828	1467	-0.964	305

**Table 3** continued

Instance	$Z_{best}$	BiF&F-f			BiF&F-s			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
08-433-8-150-200	1436	1427	0.63	5126	1436	0	2452	<b>1445</b>	-0.63	1687	<b>1443</b>	-0.487	479
09-505-8-150-200	1413	1403	0.71	987	1413	0	7027	<b>1421</b>	-0.57	2294	<b>1418</b>	-0.354	626
10-577-8-150-200	1548	1543	0.32	1763	1548	0	4201	<b>1554</b>	-0.39	2009	<b>1554</b>	-0.388	721
01-337-6-200-400	1589	1584	0.31	54	1589	0	1773	<b>1590</b>	-0.06	813	<b>1590</b>	-0.063	613
02-385-6-200-400	1600	1600	0	547	1599	0.06	2983	<b>1601</b>	-0.06	1505	<b>1603</b>	-0.188	1489
03-433-6-200-400	1621	1621	0	233	1621	0	239	<b>1622</b>	-0.06	2480	<b>1622</b>	-0.062	1435
04-481-6-200-400	1642	1641	0.06	610	1642	0	6229	1641	0.06	2647	1642	0	1793
05-529-6-200-400	1651	1651	0	396	1651	0	3683	<b>1656</b>	-0.30	4511	<b>1655</b>	-0.242	1524
06-577-6-200-400	1686	1685	0.06	696	1686	0	1393	1686	0	6748	1686	0	5138
07-361-6-200-400	1612	1608	0.25	574	1612	0	2390	<b>1614</b>	-0.12	848	<b>1613</b>	-0.062	819
08-433-6-200-400	1610	1610	0	724	1606	0.25	685	<b>1612</b>	-0.12	2700	<b>1612</b>	-0.124	3303
09-505-6-200-400	1620	1618	0.12	1923	1620	0	11069	<b>1622</b>	-0.12	1990	<b>1621</b>	-0.062	3474
10-577-6-200-400	1662	1661	0.06	6783	1662	0	12692	1662	0	6310	1662	0	3778
01-337-7-200-400	1817	1817	0	118	1817	0	120	<b>1819</b>	-0.11	1677	<b>1820</b>	-0.165	618
02-385-7-200-400	1837	1833	0.22	209	1837	0	4925	<b>1839</b>	-0.11	991	<b>1840</b>	-0.163	1465
03-433-7-200-400	1864	1864	0	456	1864	0	2600	1864	0	3805	1864	0	1241
04-481-7-200-400	1890	1890	0	6572	1889	0.05	4746	1890	0	2198	<b>1891</b>	-0.053	1995
05-529-7-200-400	1910	1910	0	4424	1910	0	7882	<b>1913</b>	-0.16	5123	<b>1913</b>	-0.157	1671
06-577-7-200-400	1947	1947	0	2518	1947	0	8065	1947	0	4125	<b>1948</b>	-0.051	2342
07-361-7-200-400	1843	1840	0.16	422	1843	0	3754	<b>1844</b>	-0.05	808	<b>1844</b>	-0.054	977
08-433-7-200-400	1848	1841	0.38	840	1848	0	10204	1848	0	1601	1850	-0.108	1365
09-505-7-200-400	1871	1871	0	8573	1870	0.05	5774	1873	-0.11	2662	1871	0	3101
10-577-7-200-400	1919	1919	0	4659	1919	0	1410	<b>1919</b>	0	6215	<b>1920</b>	-0.052	3755

**Table 3** continued

Instance	$Z_{best}$	BiF&F-f			BiF&F-s			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
01-337-8-200-400	2032	2032	0	155	2032	0	1236	<b>2039</b>	-0.34	1286	<b>2041</b>	-0.443	763
02-385-8-200-400	2065	2064	0.05	1216	2065	0	5641	<b>2066</b>	-0.05	2121	<b>2068</b>	-0.145	1078
03-433-8-200-400	2097	2096	0.05	574	2097	0	3173	2097	0	2724	<b>2098</b>	-0.048	1649
04-481-8-200-400	2127	2127	0	112	2127	0	103	<b>2130</b>	-0.14	3317	<b>2131</b>	-0.188	1627
05-529-8-200-400	2156	2156	0	1038	2155	0.05	7914	<b>2162</b>	-0.28	3764	<b>2163</b>	-0.325	2252
06-577-8-200-400	2204	2204	0	7385	2204	0	1658	2205	-0.05	6426	2204	0	2289
07-361-8-200-400	2063	2063	0	1964	2063	0	3762	<b>2063</b>	0	889	<b>2064</b>	-0.048	886
08-433-8-200-400	2070	2068	0.10	683	2070	0	3589	<b>2072</b>	-0.10	1974	<b>2074</b>	-0.193	2000
09-505-8-200-400	2115	2115	0	11022	2115	0	17380	<b>2116</b>	-0.05	2693	<b>2118</b>	-0.142	2515
10-577-8-200-400	2172	2168	0.18	3876	2172	0	15678	2171	0.05	3326	<b>2173</b>	-0.046	3197
01-337-6-345-720	2530	2530	0	136	2530	0	136	2530	0	178	2530	0	122
02-385-6-350-713	2614	2614	0	533	2614	0	752	2613	0.04	2703	2613	0.038	897
03-433-6-330-675	2520	2520	0	320	2520	0	319	2520	0	6320	2520	0	2246
04-481-6-335-713	2603	2603	0	993	2603	0	1844	2603	0	2755	2603	0	6341
05-529-6-340-705	2696	2696	0	2317	2696	0	6956	2696	0	5675	2696	0	6799
06-577-6-365-683	2932	2932	0	1798	2932	0	7050	2931	0.03	16945	2931	0.034	7305
07-361-6-335-668	2489	2489	0	183	2489	0	187	2489	0	2097	2489	0	262
08-433-6-350-675	2621	2621	0	8268	2621	0	11461	<b>2622</b>	-0.04	3121	<b>2623</b>	-0.076	3846
09-505-6-360-660	2775	2775	0	5661	2775	0	11315	<b>2780</b>	-0.18	12326	<b>2782</b>	-0.252	5697
10-577-6-375-675	2962	2960	0.07	6664	2959	0.10	10222	2962	0	8460	<b>2963</b>	-0.034	10388
01-337-7-345-720	2859	2859	0	150	2859	0	153	<b>2860</b>	-0.03	2235	<b>2860</b>	-0.035	1518
02-385-7-350-713	2965	2964	0.03	226	2965	0	652	2965	0	2853	2965	0	1946
03-433-7-330-675	2860	2860	0	283	2860	0	270	2860	0	2530	2860	0	2845

Table 3 continued

Instance	$Z_{best}$	BiF&F-f			BiF&F-s			VSS-Tabu			VSS-SA		
		$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB	$Z_{max}$	RPE	TTB
04-481-7-335-713	2954	2954	0	555	2954	0	556	2954	0	4768	2954	0	1572
05-529-7-340-705	3075	3074	0.03	2349	3075	0	6881	3074	0.03	7201	3074	0.033	6259
06-577-7-365-683	3353	3351	0.06	1840	3353	0	7411	3351	0.06	11492	3352	0.030	10810
07-361-7-335-668	2814	2812	0.07	849	2814	0	3306	2812	0.07	2429	2813	0.036	2760
08-433-7-350-675	2971	2970	0.03	916	2971	0	1425	<b>2972</b>	-0.03	5321	<b>2973</b>	-0.067	2772
09-505-7-360-660	3163	3157	0.19	13277	3163	0	13864	3162	0.03	8331	3163	0	6937
10-577-7-375-675	3381	3380	0.03	4890	3381	0	14379	<b>3383</b>	-0.06	11735	<b>3383</b>	-0.059	8549
01-337-8-345-720	3159	3159	0	927	3159	0	1745	3158	0.03	1462	3159	0	1576
02-385-8-350-713	3291	3290	0.03	144	3291	0	583	3291	0	2850	3291	0	3018
03-433-8-330-675	3190	3190	0	455	3190	0	461	3190	0	4288	3190	0	3143
04-481-8-335-713	3289	3289	0	451	3289	0	426	3289	0	283	3289	0	314
05-529-8-340-705	3434	3432	0.06	1953	3434	0	7418	3434	0	4145	3434	0	4396
06-577-8-365-683	3749	3749	0	997	3749	0	1007	3748	0.03	14090	3748	0.027	9342
07-361-8-335-668	3116	3116	0	187	3116	0	189	<b>3117</b>	-0.03	3090	<b>3117</b>	-0.032	2248
08-433-8-350-675	3302	3301	0.03	1834	3302	0	3455	3303	-0.03	5691	<b>3303</b>	-0.030	5120
09-505-8-360-660	3525	3510	0.43	2282	3525	0	14499	<b>3528</b>	-0.09	7370	<b>3529</b>	-0.113	11119
10-577-8-375-675	3783	3781	0.05	2784	3783	0	11663	3781	0.05	10154	3783	0	9865
Average			0.18	1365.11		0.02	3205.64		-0.241	2360		-0.227	1695

**Table 4** Methods improvements

	BiF&F-f	BiF&F-s	VSS-Tabu	VSS-SA
BiF&F-f	–	13	4	3
BiF&F-s	61	–	11	6
VSS-Tabu	91	76	–	27
VSS-SA	96	79	30	–

## Conclusion

The Capacitated Team Orienteering Problem is a new variant of the Vehicle Routing Problem, in which the total collected profit from the visited customers must be maximized, while adhering to all resource limitation constraints related to the capacity of the vehicles and to the predefined travel time limit. In this paper, we propose a Variable Space Search algorithm that combines two search spaces to solve CTOP. Our algorithm is based on the alternation between the *giant tours* and the *routes* search spaces in order to further improve the quality of the obtained solutions. An Adaptive Iterative Destructive/Constructive Heuristic is developed to start our solution process with high quality solutions. Several local search techniques and perturbation procedures are developed in each search space in order to improve the quality of the initial solutions constructed. Furthermore, using a Tabu Search and a Simulated Annealing procedure in our framework appears to be more efficient since they avoid being stuck in local optima. Computational results performed on the CTOP benchmark show the effectiveness and the competitiveness of our approaches since we are able to find substantial improvements with respect to the other results found in the literature, within a considerably lower computational time.

In order to further improve the performance of our VSS algorithms, additional local searches could be applied in the *routes* search space and the *giant tours* search space. Several improvements could also be applied on our AIDCH in order to provide initial solutions with better quality. Future research can extend our methodology and adopt it to solve other variants of the Vehicle Routing Problem with Profit, as the Capacitated Profitable Tour Problem (CPTP). The objective in this variant is to maximize the difference between the total collected profit and the traveled distance while respecting the limited capacity. In addition, we will work on developing exact methods in order to optimally solve CTOP.

**Acknowledgements** The work presented in this paper was achieved in the framework of the Labex MS2T, funded by the French Government, via the program “Investments for the future” managed by the National Agency for Research (Reference ANR-11-IDEX-0004-02). It was also partially supported by the TCDU project (Collaborative Transportation in Urban Distribution, ANR-14-CE22-0017).

## References

- Affi, S., Dang, D.C., Moukrim, A.: Heuristic solutions for the vehicle routing problem with time windows and synchronized visits. *Optim. Lett.* **10**(3), 511–525 (2016)

- Archetti, C., Feillet, D., Hertz, A., Speranza, M.G.: The capacitated team orienteering and profitable tour problems. *J. Oper. Res. Soc.* **60**(6), 831–842 (2009)
- Archetti, C., Bianchessi, N., Speranza, M.G.: Optimal solutions for routing problems with profits. *Discrete Appl. Math.* **161**(4), 547–557 (2013)
- Ben-Said, A., El-Hajj, R., Moukrim, A.: An adaptive heuristic for the Capacitated Team Orienteering Problem. In: IFAC-PapersOnLine 8th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2016 49(12):1662–1666 (2016)
- Boussier, S., Feillet, D., Gendreau, M.: An exact algorithm for team orienteering problems. *4OR* **5**(3), 211–230 (2007)
- Butt, S.E., Cavalier, T.M.: A heuristic for the multiple tour maximum collection problem. *Comput. Oper. Res.* **21**(1), 101–111 (1994)
- Chao, I.M., Golden, B.L., Wasil, E.A.: A fast and effective heuristic for the orienteering problem. *Eur. J. Oper. Res.* **88**(3), 475–489 (1996)
- Christofides, N., Mingozzi, A., Toth, P.: The Vehicle Routing Problem. In: Christofides, N., Mingozzi, A., Toth, P., Sandi, C. (eds.) *Combinatorial Optimization*, pp. 315–338. Wiley, Chichester (1979)
- Dang, D.C., Guibadj, R.N., Moukrim, A.: An effective PSO-inspired algorithm for the team orienteering problem. *Eur. J. Oper. Res.* **229**(2), 332–344 (2013)
- Duhamel, C., Lacomme, P., Prins, C., Prodhon, C.: A GRASP×ELS approach for the capacitated location-routing problem. *Comput. Oper. Res.* **37**(11), 1912–1923 (2010)
- Duhamel, C., Lacomme, P., Quilliot, A., Toussaint, H.: A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Comput. Oper. Res.* **38**(3), 617–640 (2011)
- Feillet, D., Dejax, P., Gendreau, M.: Traveling salesman problems with profits. *Transp. Sci.* **39**(2), 188–205 (2005)
- Gendreau, M., Laporte, G., Semet, F.: A tabu search heuristic for the undirected selective travelling salesman problem. *Eur. J. Oper. Res.* **106**(2–3), 539–545 (1998)
- Hertz, A., Plumettaz, M., Zufferey, N.: Variable space search for graph coloring. *Discrete Appl. Math.* **156**(13), 2551–2560 (2008)
- Jepsen, M.K., Petersen, B., Spoorendonk, S., Pisinger, D.: A branch-and-cut algorithm for the capacitated profitable tour problem. *Discrete Optim.* **14**(Supplement C), 78–96 (2014)
- Kim, B.I., Li, H., Johnson, A.L.: An augmented large neighborhood search method for solving the team orienteering problem. *Expert Syst. Appl.* **40**(8), 3065–3072 (2013)
- Lacomme, P., Toussaint, H., Duhamel, C.: A GRASP×ELS for the vehicle routing problem with basic three-dimensional loading constraints. *Eng. Appl. Artif. Intell.* **26**(8), 1795–1810 (2013)
- Luo, Z., Cheang, B., Lim, A., Zhu, W.: An adaptive ejection pool with toggle-rule diversification approach for the capacitated team orienteering problem. *Eur. J. Oper. Res.* **229**(3), 673–682 (2013)
- Pirkwieser, S., Raidl, G.R.: Multilevel variable neighborhood search for periodic routing problems. In: 10th European Conference on Evolutionary Computation in Combinatorial Optimization, EvoCOP, pp. 226–238 (2010)
- Tarantilis, C., Stavropoulou, F., Repoussis, P.: The capacitated team orienteering problem: a bi-level filter-and-fan method. *Eur. J. Oper. Res.* **224**(1), 65–78 (2013)
- Yamada, T., Kataoka, S., Watanabe, K.: Heuristic and exact algorithms for the disjunctively constrained knapsack problem. *Inf. Process. Soc. Jpn. J.* **43**(9), 2864–2870 (2002)
- Zhang, Z., Che, O., Cheang, B., Lim, A., Qin, H.: A memetic algorithm for the multiperiod vehicle routing problem with profit. *Eur. J. Oper. Res.* **229**(3), 573–584 (2013)