LARGE NEIGHBORHOODS WITH IMPLICIT CUSTOMER SELECTION FOR PRIZE-COLLECTING VEHICLE ROUTING AND TEAM-ORIENTEERING PROBLEMS

Thibaut Vidal
LIDS – Massachusetts Institute of Technology
77 Massachusetts Avenue, Room 32-D566, Cambridge, MA 02139, United States
Thibaut.Vidal@cirrelt.ca

Nelson Maculan
COPPE – Federal University of Rio de Janeiro, Brazil

Luis Satoru Ochi, Puca Huachi Vaz Penna
Instituto de Computação – Fluminense Federal University, Niterói, Brazil

ABSTRACT

In the Team Orienteering Problem (TOP), we are given geographically-scattered customers associated with rewards. A fixed number of distance-constrained routes must be designed to visit a subset of customers while maximizing the total profit. This problem is linked with numerous applications in transportation and logistics, e.g. for customers selection, oil-field exploitation, humanitarian relief, or military surveillance. We propose here new neighborhood searches exploring an exponential number of solutions in pseudo-polynomial time. The search is conducted on "exhaustive" solutions visiting all customers, while an efficient *Select* algorithm, based on resource-constrained shortest paths, is repeatedly used for selecting the customers to be serviced and evaluating the routes. Extensive computational experiments demonstrate the notable contribution of these neighborhood structures inside a local search and iterated local search. The simplest local search, stopping at the first local optimum, reaches an average gap of 0.09% on classic TOP instances, matching or outperforming the current best metaheuristics.

KEYWORDS. Vehicle Routing, Team Orienteering, Local Search, Large Neighborhoods
Main areas: Logistics and Transportation, Combinatorial Optimization

RESUMO

No *Team Orienteering Problem* (TOP), os clientes geograficamente dispersos estão associados a prêmios. Um número fixo de rotas com restrições de distância devem ser determinadas para visitar um subconjunto de clientes, de modo a maximizar o prêmio total. Este problema está relacionado com inúmeras aplicações em transporte e logística, por exemplo, seleção de clientes, exploração de campos de petróleo, ajuda humanitária ou vigilância militar. Neste trabalho são propostas novas vizinhanças que exploram um número exponencial de soluções em tempo pseudo-polinomial. Esta busca é realizada de forma exaustiva, visitando todos os clientes da solução, enquanto um algoritmo *Select* eficiente, baseado no caminho mais curto com restrições de recursos, é usado repetidamente para selecionar os clientes a serem atendidos e avaliar as rotas. Experimentos computacionais demonstram a notável contribuição destas estruturas de vizinhança integradas em uma Busca Local e em uma heurística *Iterated Local Search*. A busca local simples, parando no primeiro ótimo local, consegue obter um desvio médio de apenas 0,09% em instâncias clássicas do TOP, igualando ou superando as melhores metaheurísticas da literatura.

PALAVRAS-CHAVE. Roteamento de Veículos, Team Orienteering, Busca Local
Áreas Principais: L&T- Logística e Transporte, OC - Otimização Combinatória.

1 Introduction

Vehicle Routing Problems (VRP) with Profits seek to select a subset of customers, each one being associated with a reward, and design at most m vehicle itineraries starting and ending at a central depot to visit them. These problems have been the focus of extensive research, as illustrated by the surveys of Feillet *et al.* (2005) and Vansteenwegen *et al.* (2010), because of their difficulty and their numerous practical applications in logistics (Hemmelmayr *et al.*, 2009; Tricoire *et al.*, 2010), manufacturing (Tang e Wang, 2006), robotics (Falcon *et al.*, 2012), humanitarian relief (Campbell *et al.*, 2008) and military reconnaissance (Mufalli *et al.*, 2012), among others.

Three main settings are usually considered in the vehicle routing literature: profit maximization under distance constraints, called Team Orienteering Problem (TOP, Chao *et al.* 1996), maximization of profit minus travel costs under capacity constraints, called Capacitated Profitable Tour Problem (CPTP, Archetti *et al.* 2009), and the so-called VRP with Private Fleet and Common Carrier (VRPPFCC, Bolduc *et al.* 2008) in which customers can be delegated to an external logistics provider, subject to a cost. For the sake of conciseness, the scope of this paper will remain limited to the TOP. Still, the proposed methodology applies for the three problems.

To address the team-orienteering problem, we propose to conduct the search on an "exhaustive" solution representation, which only specifies the assignment and sequencing of all customers to vehicles. For each such exhaustive solution, a *Select* algorithm, based on a Resource-Constrained Shortest Path is used to perform the optimal selection of customers and evaluate the route costs. We then introduce a new Combined Local Search (CLS) working on this solution representation. The resulting local search explores an exponential set of Prize-Collecting VRP solutions, obtained from one standard VRP move combined with a exponential possible combinations of insertions and removals of customers, in pseudo-polynomial time. The contributions of this work are the following. 1) A new large neighborhood is introduced for the TOP. 2) Pruning and re-optimization techniques are proposed to perform an efficient search. 3) These neighborhoods are tested within two heuristic frameworks, a local-improvement procedure and an iterated local search. 4) Even the simple local-improvement procedure built on this neighborhood demonstrates outstanding performances on extensively-studied TOP benchmark instances, matching or outperforming all current problem-tailored metaheuristics in the literature.

2 Problem statement and mathematical formulation

Let $G = (\mathcal{V}, \mathcal{E})$ be a complete undirected graph with $|\mathcal{V}| = n + 1$ nodes. Node $v_0 \in \mathcal{V}$ represents a depot, where a fleet of m identical vehicles is based. The other nodes $v_i \in \mathcal{V} \backslash \{v_0\}$, for $i \in \{1, \dots, n\}$, represent the customers, characterized by a profit p_i. Without loss of generality, $p_0 = 0$. Edges $(i, j) \in \mathcal{E}$ represent the possibility of traveling directly from a node $v_i \in \mathcal{V}$ to $v_j \in \mathcal{V}$ for a distance/duration d_{ij}. The objective of the TOP is to find a set of $|\mathcal{R}| \leq m$ or less vehicle routes, i.e. cycle $\sigma = (\sigma(1), \dots, \sigma(|\sigma|) \in \mathcal{R}$ starting and ending at the depot such that each customer is serviced at most one time. For any route, the sum of traveled distance must be smaller than D_{max} and the total collected prize over all routes must be maximized.

A mathematical formulation of the TOP is given in Equations (1-8). This model is based on two families of binary variables, y_{ik}, designating the assignment of customer i to vehicle k by the value 1 (and 0, otherwise; $y_{0k} = 1$ signals vehicle k operates), and x_{ijk}, taking the value 1 when vehicle k visits node v_j immediately after node v_i ($i \neq j$).

$$\text{Maximize} \sum_{i=0}^{n} \sum_{k=1}^{m} p_i y_{ik} \tag{1}$$

$$\text{Subject to} \sum_{k=1}^{m} y_{ik} \leq 1 \qquad i = 1, \dots, n \tag{2}$$

$$\sum_{k=1}^{m} y_{0k} \leq m \tag{3}$$

$$\sum_{i=0}^{n}\sum_{j=0}^{n} d_{ij}x_{ijk} \leq D_{max} \qquad k = 1, \ldots, m \tag{4}$$

$$\sum_{j=0}^{n} x_{ijk} = \sum_{j=0}^{n} x_{jik} = y_{ik} \qquad i = 0, \ldots, n;\ k = 1, \ldots, m \tag{5}$$

$$\sum_{v_i \in S}\sum_{v_j \in S} x_{ijk} \leq |S| - 1 \qquad k = 1, \ldots, m; S \in V\backslash\{0\}; |S| \geq 2 \tag{6}$$

$$y_{ik} \in \{0, 1\} \qquad i = 0, \ldots, n;\ k = 1, \ldots, m \tag{7}$$

$$x_{ijk} \in \{0, 1\} \qquad i = 0, \ldots, n;\ j = 0, \ldots, n;\ k = 1, \ldots, m \tag{8}$$

Constraints (2) and (3) enforce, respectively, the assignment of each customer to a single vehicle and the maximum number of vehicles operating out of the depot. Constraint (4) limits the maximum duration of a route. Constraints (5) - (6) are related to the structure of the routes, guaranteeing the selection of an adequate number of arcs entering into and exiting from each node (depot and customers), and eliminating sub-tours.

3 Related literature

Prize collecting VRPs have been the subject of a well-developed literature since the 1980s. The three problems mentioned in the introduction are NP-hard. The current exact methods (Boussier *et al.*, 2007; Archetti *et al.*, 2013) can solve some instances with up to 200 customers, but mostly when the number of visited customers in the optimal solution remains rather small (less than 30). Heuristics are currently the method of choice for larger problems.

TOP heuristics and metaheuristics have been the subject of a large attention in the past years, perhaps due to the rapid availability of common benchmark instance (Chao *et al.*, 1996). A wide range of metaheuristic frameworks have been investigated, neighborhood-based methods (Vidal *et al.*, 2013) tending to be privileged over population-based search. Tang e Miller-Hooks (2005) proposed a tabu search with adaptive memory, profiting from feasible and infeasible solutions in the search process. Archetti *et al.* (2006) introduced a rich family of metaheuristics based on tabu or variable neighborhood search. Ke *et al.* (2008) developed ant-colony optimization techniques, and study four alternative, sequential, deterministic-concurrent, random-concurrent, and simultaneous, approaches for constructing new solutions. Bouly *et al.* (2009) introduced a hybrid GA based on giant-tour solution representation, which is hybridized later on with PSO in Dang *et al.* (2011). Vansteenwegen *et al.* (2009) proposed a guided local search, and a path relinking approach is presented in Souffriau *et al.* (2010). Finally, a multi-start simulated annealing method is introduced in Lin (2013).

Several other VRP with profits have been addressed in the literature, notably the profitable tour problem (Archetti e Speranza, 2008), the VRP with private fleet and common carrier (Potvin e Naud, 2011; Stenger *et al.*, 2012, among others), and the TOP with time windows (Labadie *et al.*, 2012; Lin e Yu, 2012, among others). Similarly to a wide majority of VRP publications, recent research has been for the most part focused on finding more sophisticated metaheuristic strategies, rather than improving the low-level neighborhood structures, which have been the same for many years. The goal of our paper is to break with this common practice by introducing a new family of combined neighborhoods. These neighborhoods can be subsequently applied in any metaheuristic framework, in possible cooperation with existing neighborhood structures.

4 Proposed neighborhoods

All the previously-mentioned efficient metaheuristics rely on local-search improvement procedures to achieve high-quality solutions. The most commonly-used neighborhoods include separate moves for changing the selection of customers with INSERT, REMOVE or REPLACE moves, and changing the assignment and sequencing of customer visits with SWAP, RELOCATE, 2-OPT, 2-OPT* or CROSS with is equivalent to 4-OPT*. We refer to Feillet *et al*. (2005) and Vidal *et al*. (2013) for a description of classic neighborhoods. However, neighborhoods which consider separately the changes of selection and sequencing/assignment may overlook a wide range of simple solution improvements. Especially, combined INSERT or REMOVE, in the same route, with a change of sequencing in route r such as RELOCATE.

4.1 Implicit customer selection

In this paper, we introduce a new neighborhood of exponential-size which can be searched in pseudo-polynomial time. Two main concepts are exploited: an exhaustive solution representation, and an implicit selection of customers. Prize-collecting vehicle routing problems indeed involve three families of decisions: a *selection* of customers to be visited, the *assignment* of selected customers, and the *sequencing* of customers for each vehicle. In an exhaustive solution, only the assignment to vehicles and sequencing **of all customers** are specified, without considering whether they are selected or not. This representation is identical to a complete VRP solution. Some routes may exceed the resource constraints, and some may not be profitable, e.g. off-centered customers with small profits are included.

To retrieve a VRPP solution from an exhaustive solution, a *Select* algorithm based on a resource-constrained shortest path is applied on each route. *Select* retrieves the optimal subsequence of customers, fulfilling the resource constraints, while maximizing the profits. The rationale of this overall methodology comes from the fact that sequences of non-activated deliveries will be represented in the solution at promising places, and thus can become implicitly activated by the *Select* procedure when a modification, e.g. sequence change in a local search, is operated.

For any route $\sigma \in \mathcal{R}$, the selection subproblem is formulated as a resource-constrained longest path on an auxiliary directed graph $\mathcal{H} = (\mathcal{V}, \mathcal{A})$. This methodology is illustrated in Figure 1 for a exhaustive solution with two routes. $\mathcal{V}$ contains the $|\sigma|$ nodes visited by the route. Each arc $(i, j) \in \mathcal{A}$ for $i < j$ is associated with a resource consumption $d_{\sigma(i)\sigma(j)}$, and each node $i \in \mathcal{V}$ is associated with a profit $p_{\sigma(i)}$. The goal is to find a path in this graph, respecting the resource constraint D_{max}, and maximizing the total profit.

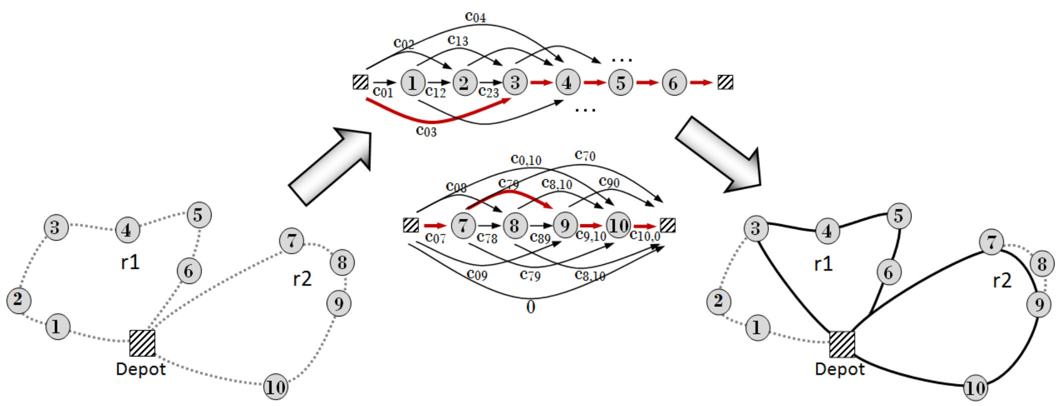

Figure 1: From an exhaustive solution to a CPTP solution

Even for an euclidean TOP, the resulting class of resource-constrained shortest path sub-problems is NP-hard. However, these problems can be solved efficiently by dynamic programming (Irnich e Desaulniers, 2005). Define a label $s = (s^{\mathrm{R}}, s^{\mathrm{P}})$ as a couple (resource,profit). To each node k, associate a set of labels $\mathcal{S}_k$, starting with $\mathcal{S}_1 = \{(0,0)\}$ for the node associated to the depot. For any $i \in \{1, \dots, |\sigma| - 1\}$, build a set of labels $\mathcal{S}'_{i+1}$ by considering any edge $(j, i+1)$ and extending all labels of j as in Equation (9).

$$\mathcal{S}'_{i+1} = \bigcup_{j|(j,i)\in\mathcal{A}} \bigcup_{s_j \in \mathcal{S}_j} (s_j^{\mathrm{R}} + d_{\sigma(j)\sigma(i)}, s_j^{\mathrm{P}} + p_{\sigma(j)})) \tag{9}$$

Any infeasible labels $s \in \mathcal{S}'_{i+1}$, such that $s^{\mathrm{R}} + d_{\sigma(i+1)\sigma(0)} > D_{max}$ is pruned from $\mathcal{S}'_{i+1}$. Indeed, distances are supposed to satisfy the triangle inequality, thus the resource consumption to return to v_0 after $\sigma(i+1)$ is greater or equal than $r_{\sigma(i+1)\sigma(0)}$, leading to this stronger feasibility bound. All dominated labels of $\mathcal{S}'_{i+1}$, i.e. label $(s^{\mathrm{R}}, s^{\mathrm{P}}) \in \mathcal{S}'_{i+1}$ such that there exists $(\bar{s}^{\mathrm{R}}, \bar{s}^{\mathrm{P}}) \in \mathcal{S}'_{i+1}$ with $s^{\mathrm{R}} \geq \bar{s}^{\mathrm{R}}$ and $s^{\mathrm{P}} \leq \bar{s}^{\mathrm{P}}$ are finally removed to yield $\mathcal{S}_{i+1}$. The feasible label with the best profit at node $|\sigma|$ gives the final cost of the TOP route, denoted as $Z^{\mathrm{TOP}}(\sigma)$. The resulting *Select* algorithm is pseudo-polynomial, with a complexity of $O(n^2 b)$, where b is an upper bound on the number of labels per node. Yet in practice, during our experiments on a large variety of benchmark instances, the number of labels remains usually very small, and never reached a value greater than 10.

4.2 Neighborhood search on the exhaustive solution

Building upon this exhaustive solution representation and the *Select* algorithm, we propose a local search procedure considering large neighborhoods with an exponential number of customer insertion and removals. This method **applies classic VRP moves** such as RELOCATE, SWAP or 2-OPT **on the exhaustive solution representation**. Evaluating the profitability of any such move requires to use the *Select* algorithm on each modified route to find the updated optimal selection. As such, insertions and removals are implicitly managed during the selection process instead of being explicitly considered by the local search, and any number of combined INSERT or REMOVE moves can be operated.

In our implementation of this technique, we consider the standard VRP neighborhoods 2-OPT, 2-OPT*, RELOCATE, SWAP and K-CROSS, limited to sequences of $K \leq 2$ customers. Each such move requires applying a maximum of two times (one for each route) the *Select* procedure. Moves are considered in random order, any improvement being directly applied (first acceptance). The method stops whenever all moves have been tried without success.

i	$d_{0,i}$	$d_{i-1,i}$	p_i
1	15	–	10
2	25	30	15
3	15	20	15
4	15	20	10
5	20	25	12
6	15	10	15
7	20	15	15
8	25	15	12
9	25	20	15
10	15	35	15

$D_{max} = 100$
$d_{7,9} = 25$
all other distances = $+\infty$

σ	$D(\sigma)$	$P(\sigma)$
(3,4,5,6)	85	52
(7,9,10)	95	45
(1,2,3,4)	100	50
(6,7,8,9)	90	57

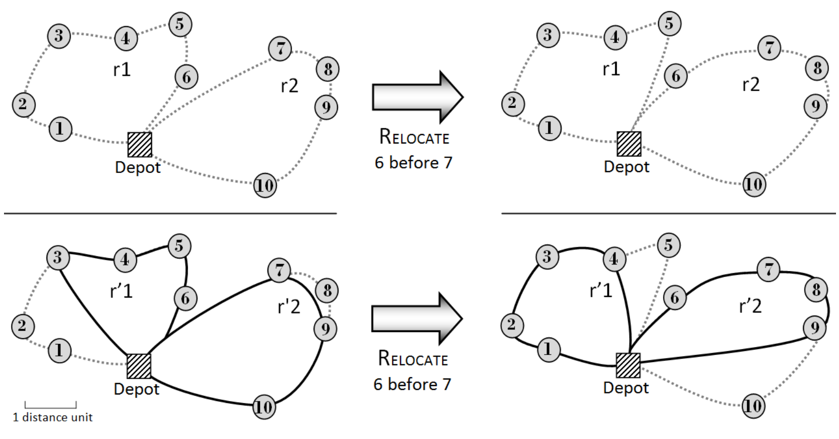

Figure 2: RELOCATE move on the exhaustive solution representation, and impact on the associated VRPP solution

Figure 2 illustrates a simple RELOCATE move applied on the exhaustive solution representation (top of the figure), and its possible impact on the associated VRPP solution (bottom of the figure). A RELOCATE of customer C6 before C7 on the exhaustive solution representation can have a dramatic impact on the associated VRPP solution, since as a consequence of the RELOCATE, the *Select* algorithm operates different choices, here a compound REMOVAL of C_5 and C_10 and INSERT of C_1 and C_2 before C_3. As a result, a feasible solution with higher profit can be attained.

4.3 Hierarchical objective and speed-up techniques

When only the primary problem objective $Z = \max \sum_{\sigma \in \mathcal{R}} Z^{\text{TOP}}(\sigma)$ is considered, most local-search moves on customers which are currently non-activated by *Select* have no impact on the current cost. This leads a "staircase" aspect of the search space, which is usually not suitable for an efficient search. To avoid this drawback and drive the non-activated customers towards promising locations, the total distance of the current exhaustive solution is considered as a secondary objective (Equation 10). The factor ω enables to scale the objectives. In our experiments, we considered $\omega = 10^{-4}$, which sets up a hierarchy without involving numerical precision issues.

$$Z' = \max \sum_{\sigma \in \mathcal{R}} Z^{\text{SELECT}}(\sigma) + \omega \sum_{\sigma \in \mathcal{R}} \sum_{i \in \{1,\dots,|\sigma|-1\}} d_{\sigma(i)\sigma(i+1)} \tag{10}$$

As a result, even if no improving move w.r.t. the primary TOP objective can be found from an incumbent solution, the neighborhood search will re-arrange the deliveries to positions where they are more likely to be activated at a further step. This may open the way to new improvements of the main objective at a later stage, without requiring any solution deterioration.

In addition, solving from scratch each such resource-constrained shortest path leads to computationally expensive move evaluations in $O(n^2 b)$. Pseudo-quadratic complexity is not acceptable in recent neighborhood-based heuristic searches which rely on a considerable number of route evaluations. To reduce this complexity, we prune several edges in the shortest-path graph. For a given integer h, only the arcs (i, j), with $(i < j)$ following the condition of Equation (11) are kept. The *sparsification parameter* $h \in [0, n]$ represents a bound on the maximum consecutive number of non-activated deliveries which can arise in a VRPP solution.

$$j < i + h \text{ or } i = 0 \text{ or } j = |\sigma| \tag{11}$$

As a consequence, only a limited number of non-activated deliveries can be located between two consecutive selected deliveries to customers. The number of non-activated deliveries located just after or just before a depot remains unlimited, thus guaranteeing the existence of a feasible solution. This limitation enables to reduce the number of edges of the auxiliary graph from $O(n^2)$ down to $O(hn)$, and dramatically reduces the complexity of solving the underlying resource-constrained shortest paths. Finally, to speed up further the resolution we rely on bi-directional dynamic programming techniques (Righini e Salani, 2006) and keep in memory the partial label trees generated during each move evaluation. These trees can subsequently be re-used to evaluate closely related moves. Standard static move pruning techniques (granular search Toth e Vigo 2003) are also exploited, and thus local-search moves between customers i and j are only evaluated if j is among the $|\Gamma| = 20$ closest vertices to i.

5 Heuristic and metaheuristic framework

The contribution of this new neighborhood is assessed by extensive computational experiments with two types of heuristics: a simple Multi-Start Local-Improvement (MS-LI) procedure, and a Multi-Start Iterated Local Search (MS-ILS). The local-improvement procedure is based on classic vehicle routing neighborhoods: 2-OPT, 2-OPT* as well as CROSS and K-CROSS exchanges with $K = 2$. Moves are explored in random order, any improving move being directly applied until no improvement can be found in the whole neighborhood, the resulting solution being a local optimum of the proposed large neighborhood. The local-improvement procedure is repeated

100 times from randomly generated initial solutions. The best of all local optima constitutes the MS-LI solution.

MS-ILS, depicted in Algorithm 1, starts from a randomly generated initial solution, then from each incumbent solution generates n_{C} child solutions by applying a shaking operator and the local-improvement procedure, the best child solution being taken as new incumbent solution for the next iteration. As in Prins (2009), shaking consists in swapping two random deliveries on the giant-tour solution representation which does not mention visits to the depot. A Split algorithm is then applied to optimally insert the depots. The method is restarted n_{P} times, each run being terminated after n_{I} consecutive iterations without improvement of the best solution. The overall best solution is finally returned.

Algorithm 1 Multi-Start Iterated Local Search

for $i_{\text{R}} = 1$ to n_{P} **do**
  $s_{\text{CURR}} \leftarrow$ getInitialSolution()
  $s_{\text{BEST}} \leftarrow s_{\text{CURR}}$
  $i_{\text{ILS}} \leftarrow 0$
  while $i_{\text{ILS}} < n_{\text{I}}$ **do**
    $S_{\text{CHILDREN}} \leftarrow$
    for $i_{\text{C}} = 1$ to n_{C} **do**
      $s \leftarrow$Shaking(s_{CURR})
      $s \leftarrow$Split(s)
      $S_{\text{CHILDREN}} \leftarrow$localImprovement($s$)
    $s_{\text{CURR}} \leftarrow$bestElement($S_{\text{CHILDREN}}$)
    if cost(s_{CURR}) $<$ cost(s_{BEST}) **then** $s_{\text{BEST}} \leftarrow s_{\text{CURR}}$; $i_{\text{ILS}} = 0$
    else $i_{\text{ILS}} = i_{\text{ILS}} + 1$
return bestSolutionEver()

6 Experimental Analyses

The performance of these two heuristics built on our new compound neighborhoods is assessed by means of extensive experiments on classic benchmark instances for the TOP of Chao *et al.* (1996). The considered instances are classified into seven sets of instances, Sets 1 to 7, which include respectively 32, 21, 33, 100, 66, 64 and 102 customers. Each instance set is declined into individual instances with between 2 to 4 vehicles and different duration limits. We focus our discussion on the larger sets, 4 to 7, since the proposed methods systematically reach the optimal solutions on the smaller instances. In addition, instances for which all methods from the recent literature find the optimal solutions have been excluded from the experiments, to only keep the 157 most difficult ones as in Souffriau *et al.* (2010)

After some preliminary tests, the parameters of MS-ILS have been set to $(n_{\text{P}}, n_{\text{I}}, n_{\text{C}}) = (5, 10, 3)$ which allows to compare to other methods in similar CPU time while keeping a good balance between restarts (n_{P}) for diversity, and search intensity (n_{I} and n_{C}). The sparsification parameter has been set to $h = 3$, a value which provides a good trade-off between quality of solution and computational effort. In the following, our methods are compared to the best current metaheuristics in the TOP literature:

- CGW : Tabu Search of Chao *et al.* (1996) – as reported by Tang e Miller-Hooks (2005)
- TMH : Tabu Search of Tang e Miller-Hooks (2005)
- GTF : Tabu Search with feasible strategy of Archetti *et al.* (2006)
- SVF : Slow variable neighborhood search of Archetti *et al.* (2006)
- ASe : Sequential Ant colony optimization of Ke *et al.* (2008)
- GLS : Guided Local Search of Vansteenwegen *et al.* (2009)

- SVNS : Skewed Variable Neighborhood Sarch of Vansteenwegen e Souffriau (2009)
- SPR : Slow Path Relinking of Souffriau *et al*. (2010)
- MSA : Multi-Start Simulated Annealing of Lin (2013)

Table 1 compares the average results of MS-LI, MS-ILS with recent state-of-the-art methods on Sets 4 to 7. In accordance with the current practice for this problem, the best solution quality out of 10 independent runs is reported for each method and problem set. This quality is expressed as a Gap (%) to the best profit ever found in the literature, averaged on all instances of the same set. The average CPU time per instance, for each method, as well as the type of processor is also reported. To further compare the performance of the methods on separate instances, Tables 2 and 3 display the detailed results of each method on the two largest sets. The previous best known solutions (BKS) from the literature are indicated in the last column. Further detailed results on all benchmark instances are available upon request.

Table 1: Summary of results on TOP benchmark instances

	CGW	**TMH**	**GTF**	**SVF**	**ASe**	**GLS**	**SVNS**	**SPR**	**MSA**	**MS-ILS**	**MS-LI**
Set 4 Gap	4.36%	1.99%	0.48%	0.06%	0.30%	2.96%	1.46%	0.11%	0.06%	**0.05%**	0.09%
n=100 T(s)	796.70	105.30	22.50	11.40	32.00	37.10	36.70	367.40	81.00	301.54	76.72
Set 5 Gap	1.36%	1.38%	**0.01%**	0.03%	0.04%	2.39%	0.61%	0.05%	**0.01%**	**0.01%**	**0.01%**
n=66 T(s)	71.30	69.50	34.20	3.50	15.10	17.40	11.20	119.90	6.60	193.97	11.31
Set 6 Gap	0.37%	0.79%	0.04%	**0.00%**	**0.00%**	1.78%	0.52%	**0.00%**	**0.00%**	**0.00%**	**0.00%**
n=64 T(s)	45.70	66.30	8.70	4.30	14.10	16.10	9.00	89.60	1.40	138.25	6.86
Set 7 Gap	2.68%	1.15%	0.29%	0.06%	**0.00%**	3.07%	1.31%	0.04%	0.03%	**0.00%**	0.07%
n=102 T(s)	432.60	160.00	10.30	12.10	24.60	30.40	27.30	272.80	32.20	309.62	50.22
CPU	SUN4/370	DECAlpha	P4 2.8G	P4 2.8G	P4 3.0G	P4 2.8G	P4 2.8G	Xe 2.5G	C2 2.5G	Xe 3.0G	Xe 3.0G

These results demonstrate the good performance of both methods. For Sets 5 and 6, the proposed methods attain a gap of less than $+0.01\%$, while on the larger instances, gap values do not exceed $+0.09\%$ for MS-LI and $+0.05\%$ for MS-ILS. It is remarkable that this simple multi-start local-improvement procedure matches or improves upon all currently existing (and ofter very intricate) metaheuristics. This illustrates the notable contribution of the new large neighborhoods, which drive the search towards local optimum of much higher quality. Run times are of the same order of magnitude as the other recent approaches, with an average of 4 minutes for MS-ILS, and 36 seconds for MS-LI. This CPU time is suitable for most operational optimization contexts.

Considering the individual results on the largest instances of Sets 4 and 7, MS-LI attains the largest number BKS, with 46/54 BKS on Set 4 and 44/44 BKS on Set 7. For these sets, the second methods yielding the most BKS values are SVF on Set 4 with 42/54, and ASe on Set 7 with 40/44. In addition, one new best solution has been found for the first time on instance p.7.3.t with a value of 1120. Given the considerable effort put on these instances during the past years, it is a major achievement. Finally, for Sets 1, 2, 3, 5, 6 and 7, all best known solutions have been found by MS-ILS. MS-LI also performs impressively well in terms of number of BKS, and is ranked among the top three methods w.r.t. this criteria on all sets.

We finally conduct a group of two-tailed paired-samples t-tests to investigate for each method X the hypothesis that "MS-ILS yields results which are significantly different than X", with the results of Set 4 and 7 expressed as percentage of deviation from the BKS for each instance. The results of these statistical tests are displayed in Table 4. At 0.05 confidence level, MS-ILS returns significantly different results than CGW, TMH, GTF, ASe, GLS, SVNS, SPR, and MS-LS. For SVF and MSA, p-values of 0.09 and 0.13 are obtained, respectively. This corroborates our conjecture that the proposed method outperforms all past algorithms. Further independent tests will be conducted to validate this statement.

Table 2: Results on TOP benchmark instances, Set 4

Inst	n	m	**CGW**	**TMH**	**GTF**	**SVF**	**ASe**	**GLS**	**SVNS**	**SPR**	**MSA**	**MS-ILS**	**MS-LI**	BKS
p4.2.a	100	2	194	202	**206**	**206**	**206**	**206**	202	**206**	**206**	**206**	**206**	206
p4.2.b	100	2	**341**	**341**	**341**	**341**	**341**	303	**341**	**341**	**341**	**341**	**341**	341
p4.2.c	100	2	440	438	**452**	**452**	**452**	447	**452**	**452**	**452**	**452**	**452**	452
p4.2.d	100	2	**531**	517	**531**	**531**	**531**	526	528	**531**	**531**	**531**	**531**	531
p4.2.e	100	2	580	593	613	**618**	**618**	602	593	**618**	**618**	**618**	**618**	618
p4.2.f	100	2	669	666	676	**687**	**687**	651	675	**687**	**687**	**687**	**687**	687
p4.2.g	100	2	737	749	756	**757**	**757**	734	750	**757**	**757**	**757**	**757**	757
p4.2.h	100	2	807	827	820	**835**	827	797	819	**835**	**835**	**835**	**835**	835
p4.2.i	100	2	858	915	899	**918**	**918**	826	916	**918**	**918**	**918**	**918**	918
p4.2.j	100	2	899	914	962	962	**965**	939	962	**965**	962	962	962	965
p4.2.k	100	2	932	963	1013	**1022**	**1022**	994	1007	**1022**	**1022**	**1022**	1010	1022
p4.2.l	100	2	1003	1022	1058	**1074**	1071	1051	1051	**1074**	1073	1071	**1074**	1074
p4.2.m	100	2	1039	1089	1098	**1132**	1130	1051	1051	**1132**	**1132**	**1132**	**1132**	1132
p4.2.n	100	2	1112	1150	1171	**1174**	1168	1117	1124	1173	**1174**	1172	1172	1174
p4.2.o	100	2	1147	1175	1192	**1218**	1215	1191	1195	**1218**	1217	**1218**	**1218**	1218
p4.2.p	100	2	1199	1208	1239	1241	**1242**	1214	1237	**1242**	1241	1241	1241	1242
p4.2.q	100	2	1242	1255	1255	1263	1263	1248	1239	1263	1259	**1265**	**1265**	1265
p4.2.r	100	2	1199	1277	1283	1285	1288	1267	1279	1286	**1290**	1281	1285	1290
p4.2.s	100	2	1286	1294	1299	1301	**1304**	1286	1295	1296	1300	1297	1301	1304
p4.2.t	100	2	1299	**1306**	**1306**	**1306**	**1306**	1294	1305	**1306**	**1306**	**1306**	**1306**	1306
p4.3.c	100	3	191	192	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**	**193**	193
p4.3.d	100	3	333	333	**335**	**335**	**335**	**335**	331	**335**	**335**	**335**	**335**	335
p4.3.e	100	3	432	465	**468**	**468**	**468**	444	460	**468**	**468**	**468**	**468**	468
p4.3.f	100	3	552	**579**	**579**	**579**	**579**	564	556	**579**	**579**	**579**	**579**	579
p4.3.g	100	3	623	646	652	**653**	**653**	644	651	**653**	**653**	**653**	**653**	653
p4.3.h	100	3	717	709	727	**729**	720	706	718	**729**	**729**	**729**	**729**	729
p4.3.i	100	3	798	785	806	**809**	796	806	807	**809**	**809**	**809**	**809**	809
p4.3.j	100	3	829	860	858	**861**	**861**	826	854	**861**	860	**861**	860	861
p4.3.k	100	3	889	906	918	**919**	918	864	902	918	**919**	**919**	**919**	919
p4.3.l	100	3	946	951	973	**979**	**979**	960	969	**979**	978	**979**	**979**	979
p4.3.m	100	3	956	1005	1049	1062	1053	1030	1047	**1063**	**1063**	**1063**	**1063**	1063
p4.3.n	100	3	1018	1119	1115	**1121**	**1121**	1113	1106	1120	**1121**	**1121**	**1121**	1121
p4.3.o	100	3	1078	1151	1157	**1172**	1170	1121	1136	1170	1170	**1172**	1170	1172
p4.3.p	100	3	1115	1218	1221	**1222**	1221	1190	1200	1220	**1222**	**1222**	**1222**	1222
p4.3.q	100	3	1222	1249	1241	1245	1252	1210	1236	**1253**	1251	**1253**	1251	1253
p4.3.r	100	3	1225	1265	1269	**1273**	1267	1239	1250	1272	1265	1272	1269	1273
p4.3.s	100	3	1239	1282	1294	**1295**	1293	1279	1280	1287	1293	**1295**	**1295**	1295
p4.3.t	100	3	1285	1288	1304	1304	**1305**	1290	1299	1299	1299	**1305**	1299	1305
p4.4.e	100	4	182	182	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**	**183**	183
p4.4.f	100	4	304	315	**324**	**324**	**324**	312	319	**324**	**324**	**324**	**324**	324
p4.4.g	100	4	460	453	**461**	**461**	**461**	**461**	**461**	**461**	**461**	**461**	**461**	461
p4.4.h	100	4	545	554	**571**	**571**	**571**	565	553	**571**	**571**	**571**	**571**	571
p4.4.i	100	4	641	627	**657**	**657**	**657**	**657**	**657**	**657**	**657**	**657**	**657**	657
p4.4.j	100	4	697	**732**	731	**732**	**732**	691	723	**732**	**732**	**732**	**732**	732
p4.4.k	100	4	770	819	816	**821**	**821**	815	**821**	**821**	**821**	**821**	**821**	821
p4.4.l	100	4	847	875	878	**880**	**880**	852	876	879	**880**	**880**	**880**	880
p4.4.m	100	4	895	910	918	918	918	910	903	**919**	**919**	**919**	916	919
p4.4.n	100	4	932	**977**	976	976	961	942	948	969	975	972	976	977
p4.4.o	100	4	995	1014	1057	**1061**	1036	937	1030	1057	**1061**	**1061**	**1061**	1061
p4.4.p	100	4	996	1056	1120	1120	1111	1091	1120	1122	**1124**	**1124**	**1124**	1124
p4.4.q	100	4	1084	1124	1157	**1161**	1145	1106	1149	1160	**1161**	**1161**	1157	1161
p4.4.r	100	4	1155	1165	1211	1207	1200	1148	1193	1213	**1216**	**1216**	1211	1216
p4.4.s	100	4	1230	1243	1256	**1260**	1249	1242	1213	1250	1256	**1260**	**1260**	1260
p4.4.t	100	4	1253	1255	**1285**	**1285**	1281	1250	1281	1280	**1285**	**1285**	**1285**	1285

7 Conclusion

A new large neighborhood concept has been introduced, leading to an efficient local-improvement technique and an iterated local search. Computational experiments demonstrate the

Table 3: Results on TOP benchmark instances, Set 7

Inst	n	m	**CGW**	**TMH**	**GTF**	**SVF**	**ASe**	**GLS**	**SVNS**	**SPR**	**MSA**	**MS-ILS**	**MS-LI**	BKS
p7.2.d	102	2	**190**	**190**	**190**	**190**	**190**	**190**	182	**190**	**190**	**190**	**190**	190
p7.2.e	102	2	275	**290**	**290**	**290**	**290**	279	289	**290**	**290**	**290**	**290**	290
p7.2.f	102	2	379	382	**387**	**387**	**387**	340	**387**	**387**	**387**	**387**	**387**	387
p7.2.g	102	2	453	**459**	**459**	**459**	**459**	440	457	**459**	**459**	**459**	**459**	459
p7.2.h	102	2	517	**521**	520	**521**	**521**	517	**521**	**521**	**521**	**521**	**521**	521
p7.2.i	102	2	576	578	579	579	**580**	568	579	**580**	579	**580**	**580**	580
p7.2.j	102	2	633	638	644	644	**646**	633	632	**646**	**646**	**646**	**646**	646
p7.2.k	102	2	693	702	**705**	**705**	**705**	691	700	**705**	**705**	**705**	**705**	705
p7.2.l	102	2	758	**767**	**767**	**767**	**767**	748	758	**767**	**767**	**767**	**767**	767
p7.2.m	102	2	811	817	824	**827**	**827**	798	**827**	**827**	**827**	**827**	824	827
p7.2.n	102	2	864	864	**888**	**888**	**888**	861	866	**888**	**888**	**888**	**888**	888
p7.2.o	102	2	934	914	**945**	**945**	**945**	897	928	**945**	**945**	**945**	**945**	945
p7.2.p	102	2	987	987	**1002**	**1002**	**1002**	954	955	**1002**	**1002**	**1002**	**1002**	1002
p7.2.q	102	2	1031	1017	1043	1043	1043	1031	1029	**1044**	1043	**1044**	**1044**	1044
p7.2.r	102	2	1082	1067	1088	**1094**	**1094**	1075	1069	**1094**	1093	**1094**	**1094**	1094
p7.2.s	102	2	1127	1116	1128	1135	**1136**	1102	1118	**1136**	1135	**1136**	**1136**	1136
p7.2.t	102	2	1173	1165	1174	**1179**	**1179**	1142	1154	1175	1172	**1179**	**1179**	1179
p7.3.h	102	3	419	416	**425**	**425**	**425**	418	**425**	**425**	**425**	**425**	**425**	425
p7.3.i	102	3	466	481	**487**	**487**	**487**	480	480	**487**	**487**	**487**	**487**	487
p7.3.j	102	3	539	563	**564**	**564**	**564**	539	543	**564**	**564**	**564**	**564**	564
p7.3.k	102	3	602	632	**633**	**633**	**633**	586	**633**	**633**	**633**	**633**	**633**	633
p7.3.l	102	3	676	681	679	683	**684**	668	681	**684**	**684**	**684**	**684**	684
p7.3.m	102	3	754	756	755	**762**	**762**	735	743	**762**	**762**	**762**	**762**	762
p7.3.n	102	3	813	789	811	813	**820**	789	804	**820**	**820**	**820**	814	820
p7.3.o	102	3	848	**874**	865	**874**	**874**	833	841	**874**	**874**	**874**	**874**	874
p7.3.p	102	3	919	922	923	927	**929**	912	918	927	927	**929**	927	929
p7.3.q	102	3	943	966	**987**	**987**	**987**	945	966	**987**	**987**	**987**	982	987
p7.3.r	102	3	1008	1011	1022	**1026**	**1026**	1015	1009	1021	**1026**	**1026**	1021	1026
p7.3.s	102	3	1064	1061	**1081**	**1081**	**1081**	1054	1070	**1081**	**1081**	**1081**	**1081**	1081
p7.3.t	102	3	1095	1098	1116	1117	1118	1080	1109	1118	1119	1120	1118	1119
p7.4.g	102	4	209	**217**	**217**	**217**	**217**	209	**217**	**217**	**217**	**217**	**217**	217
p7.4.h	102	4	283	**285**	**285**	**285**	**285**	**285**	283	**285**	**285**	**285**	**285**	285
p7.4.i	102	4	338	359	**366**	**366**	**366**	359	364	**366**	**366**	**366**	**366**	366
p7.4.k	102	4	516	503	**520**	**520**	**520**	511	518	518	**520**	**520**	**520**	520
p7.4.l	102	4	562	576	588	**590**	**590**	573	575	**590**	**590**	**590**	**590**	590
p7.4.m	102	4	610	643	**646**	**646**	**646**	638	639	**646**	**646**	**646**	**646**	646
p7.4.n	102	4	683	726	721	726	**730**	698	723	**730**	**730**	**730**	726	730
p7.4.o	102	4	728	776	778	**781**	**781**	761	778	780	**781**	**781**	**781**	781
p7.4.p	102	4	801	832	839	**846**	**846**	803	841	**846**	**846**	**846**	**846**	846
p7.4.q	102	4	882	905	898	**909**	**909**	899	896	907	**909**	**909**	**909**	909
p7.4.r	102	4	886	966	969	**970**	**970**	937	964	**970**	**970**	**970**	**970**	970
p7.4.s	102	4	990	1019	1020	**1022**	**1022**	1005	1019	**1022**	**1022**	**1022**	**1022**	1022
p7.4.t	102	4	1066	1067	1071	**1077**	**1077**	1020	1073	**1077**	**1077**	**1077**	**1077**	1077

Table 4: p-values for the paired-samples t-tests between MS-ILS and other methods

	CGW	**TMH**	**GTF**	**SVF**	**ASe**	**GLS**	**SVNS**	**SPR**	**MSA**	**MS-LS**
MS-ILS	0.0000	0.0000	0.0000	0.0916	0.0000	0.0015	0.0000	0.0031	0.1301	0.0144

remarkable performance of these neighborhoods on classic TOP benchmark instances from the literature, reaching gaps to the best known solutions off less than 0.09%. It is remarkable that a simple local-improvement method based on these rich neighborhoods reaches solutions of higher quality than many of the existing metaheuristic frameworks for this problem.

In addition to be efficient, the main strength of our approach is that it is a radically new way of designing neighborhood-search on prize collecting problems, applicable within any metaheuristic context. As such, many perspectives of research remain open to combine the two types of search spaces into a new generation of highly effective methods.

References

Archetti, C., Bianchessi, N. e Speranza, M. (2013), Optimal solutions for routing problems with profits. *Discrete Applied Mathematics*, v. 161, n. 4-5, p. 547–557.

Archetti, C., Feillet, D., Hertz, A. e Speranza, M. G. (2009), The capacitated team orienteering and profitable tour problems. *Journal of the Operational Research Society*, v. 60, n. 6, p. 831–842.

Archetti, C., Hertz, A. e Speranza, M. G. (2006), Metaheuristics for the team orienteering problem. *Journal of Heuristics*, v. 13, n. 1, p. 49–76.

Archetti, C. e Speranza, M. The split delivery vehicle routing problem: A survey. Golden, B., Raghavan, S. e Wasil, E. (Eds.), *The Vehicle Routing Problem: Latest Advances and New Challenges*, p. 103–122. Springer, New York, 2008.

Bolduc, M.-C., Renaud, J., Boctor, F. e Laporte, G. (2008), A perturbation metaheuristic for the vehicle routing problem with private fleet and common carriers. *Journal of the Operational Research Society*, v. 59, n. 6, p. 776–787.

Bouly, H., Dang, D.-C. e Moukrim, A. (2009), A memetic algorithm for the team orienteering problem. *4OR*, v. 8, n. 1, p. 49–70.

Boussier, S., Feillet, D. e Gendreau, M. (2007), An exact algorithm for team orienteering problems. *4OR*, v. 5, n. 3, p. 211–230.

Campbell, A., Vandenbussche, D. e Hermann, W. (2008), Routing for Relief Efforts. *Transportation Science*, v. 42, n. 2, p. 127–145.

Chao, I., Golden, B. e Wasil, E. (1996), The team orienteering problem. *European Journal of Operational Research*, v. 88, n. 3, p. 464–474.

Dang, D.-C., Guibadj, R. e Moukrim, A. A PSO-Based Memetic Algorithm for the Team Orienteering Problem. *Applications of Evolutionary Computation*, volume 6625 of *LNCS*, p. 471–480. Springer Berlin Heidelberg, 2011.

Falcon, R., Li, X., Nayak, A. e Stojmenovic, I. A Harmony-Seeking Firefly Swarm to the Periodic Replacement of Damaged Sensors by a Team of Mobile Robots. *ICC'12*, p. 4914–4918. ISBN 978-1-4577-2053-6, 2012.

Feillet, D., Dejax, P. e Gendreau, M. (2005), Traveling Salesman Problems with Profits. *Transportation Science*, v. 39, n. 2, p. 188–205.

Hemmelmayr, V., Doerner, K., Hartl, R. e Savelsbergh, M. (2009), Delivery strategies for blood products supplies. *OR spectrum*, v. 31, n. 4, p. 707–725.

Irnich, S. e Desaulniers, G. Shortest Path Problems with Resource Constraints. *Column Generation*, p. 33—-65, 2005.

Ke, L., Archetti, C. e Feng, Z. (2008), Ants can solve the team orienteering problem. *Computers & Industrial Engineering*, v. 54, n. 3, p. 648–665.

Labadie, N., Mansini, R., Melechovský, J. e Wolfler Calvo, R. (2012), The Team Orienteering Problem with Time Windows: An LP-based Granular Variable Neighborhood Search. *European Journal of Operational Research*, v. 220, n. 1, p. 15–27.

Lin, S.-W. (2013), Solving the team orienteering problem using effective multi-start simulated annealing. *Applied Soft Computing*, v. 13, n. 2, p. 1064–1073.

Lin, S.-W. e Yu, V. F. (2012), A simulated annealing heuristic for the team orienteering problem with time windows. *European Journal of Operational Research*, v. 217, n. 1, p. 94–107.

Mufalli, F., Batta, R. e Nagi, R. (2012), Simultaneous sensor selection and routing of unmanned aerial vehicles for complex mission plans. *Computers & Operations Research*, v. 39, n. 11, p. 2787–2799.

Potvin, J.-Y. e Naud, M.-A. (2011), Tabu search with ejection chains for the vehicle routing problem with private fleet and common carrier. *Journal of the Operational Research Society*, v. 62, n. 2, p. 326–336.

Prins, C. A GRASP - Evolutionary Local Search Hybrid for the Vehicle Routing Problem. Pereira, F. e Tavares, J. (Eds.), *Bio-Inspired Algorithms for the Vehicle Routing Problem*, p. 35–53. Springer Berlin Heidelberg, 2009.

Righini, G. e Salani, M. (2006), Symmetry helps: Bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints. *Discrete Optimization*, v. 3, n. 3, p. 255–273.

Souffriau, W., Vansteenwegen, P., Vanden Berghe, G. e Van Oudheusden, D. (2010), A Path Relinking approach for the Team Orienteering Problem. *Computers & Operations Research*, v. 37, n. 11, p. 1853–1859.

Stenger, A., Vigo, D., Enz, S. e Schwind, M. (2012), An adaptive variable neighborhood search algorithm for a vehicle routing problem arising in small package shipping. *Transportation Science*, v. 47, n. 1, p. 64–80.

Tang, H. e Miller-Hooks, E. (2005), A TABU search heuristic for the team orienteering problem. *Computers & Operations Research*, v. 32, n. 6, p. 1379–1407.

Tang, L. e Wang, X. (2006), Iterated local search algorithm based on very large-scale neighborhood for prize-collecting vehicle routing problem. *The International Journal of Advanced Manufacturing Technology*, v. 29, n. 11-12, p. 1246–1258.

Toth, P. e Vigo, D. (2003), The Granular Tabu Search and Its Application to the Vehicle-Routing Problem. *INFORMS Journal on Computing*, v. 15, n. 4, p. 333–346.

Tricoire, F., Romauch, M., Doerner, K. e Hartl, R. (2010), Heuristics for the multi-period orienteering problem with multiple time windows. *Computers & Operations Research*, v. 37, n. 2, p. 351–367.

Vansteenwegen, P. e Souffriau, W. Metaheuristics for tourist trip planning. Sörensen, K., Sevaux, M., Habenicht, W. e Geiger, M. (Eds.), *Metaheuristics in the Service Industry*, Lecture Notes in Economics and Mathematical Systems, p. 15–31. Springer-Verlag Berlin Heidelberg, 2009.

Vansteenwegen, P., Souffriau, W., Berghe, G. e Oudheusden, D. (2009), A guided local search metaheuristic for the team orienteering problem. *European Journal of Operational Research*, v. 196, n. 1, p. 118–127.

Vansteenwegen, P., Souffriau, W. e Oudheusden, D. (2010), The orienteering problem: A survey. *European Journal of Operational Research*, v. 209, n. 1, p. 1–10.

Vidal, T., Crainic, T., Gendreau, M. e Prins, C. (2013), Heuristics for Multi-Attribute Vehicle Routing Problems: A Survey and Synthesis. *European Journal of Operational Research, Forthcoming*.