Hindawi
Mathematical Problems in Engineering
Volume 2017, Article ID 9285403, 6 pages
https://doi.org/10.1155/2017/9285403

*Research Article*

# Exact Algorithm for the Capacitated Team Orienteering Problem with Time Windows

**Junhyuk Park,[1] Jongsung Lee,[2] SeHwan Ahn,[3] Jungryul Bae,[4] and Hyunchul Tae[3]**

[1] *WM Logistics, Magnolia, TX, USA*
[2] *Samsung Electronics Corporate, Suwon, Republic of Korea*
[3] *Korea Institute of Industrial Technology (KITECH), Cheonan, Republic of Korea*
[4] *Department of Mathematics, Yonsei University, Seoul, Republic of Korea*

Correspondence should be addressed to Hyunchul Tae; sage@kitech.re.kr

Received 8 July 2016; Revised 2 November 2016; Accepted 19 December 2016; Published 26 January 2017

Academic Editor: Yakov Strelniker



The capacitated team orienteering problem with time windows (CTOPTW) is a problem to determine players' paths that have the maximum rewards while satisfying the constraints. In this paper, we present the exact solution approach for the CTOPTW which has not been done in previous literature. We show that the branch-and-price (B&P) scheme which was originally developed for the team orienteering problem can be applied to the CTOPTW. To solve pricing problems, we used implicit enumeration acceleration techniques, heuristic algorithms, and *ng*-route relaxations.

## 1. Introduction

The orienteering problem (OP) [1], an NP-hard integer problem (IP), originates from a sport in which a player finds a path that has the maximum reward. The OP arises in a map with a set of points and connecting arcs. The player moves from the starting point and has to reach the ending point within a time limit. Each point has its own reward and the player can collect rewards by visiting points. Many exact and heuristic algorithms have been proposed to solve the OP [2].

The team orienteering problem (TOP) is a variant of the OP that considers multiple players. TOP is also NP-hard. Many algorithms have been proposed for the TOP, including heuristic algorithms [3–13] and exact algorithms [14–17]. Some practical problems in the field of the vehicle routing research exhibit the form of OP/TOP. For example, the home fuel delivery problem and the recruiting problem have the same problem characteristics with the OP/TOP [3]. Therefore, several vehicle routing constraints like the vehicle capacity and time windows constraints have been considered in the TOP.

The TOP with time windows (TOPTW) is a variant of the TOP that additionally considers the time window constraint; that is, a player can visit a point only within a specified interval (time window). Many heuristic algorithms [18–27] and one exact algorithm [17] have been proposed for the TOPTW.

The capacitated TOP (CTOP) is a variant of the TOP that additionally considers a capacity constraint where each point has its own demand and all players have the same capacity. A player cannot visit a set of points if the sum of the demands of the visited points exceeds the player's capacity. Our previous studies found two heuristic algorithms [28, 29] and one exact algorithm [30] for the CTOP.

The capacitated TOP with time windows (CTOPTW) is a hybrid of the CTOP and TOPTW. The CTOPTW was first studied in [31] and the authors proposed the iterated local search (ILS) heuristic algorithm. Later, [32–35] proposed improving algorithms but proposed no exact algorithm for the CTOPTW. To the best of our knowledge, our paper is the first to present the exact solution of the CTOPTW.

We apply the branch-and-price (B&P) scheme of Boussier et al. [16] to the CTOPTW. The B&P scheme was originally developed for the TOP, but it can be applied to the CTOPTW without any modification. We develop implicit enumeration to solve pricing problems that arise in the B&P for the CTOPTW.

The remainder of this paper is organized as follows. Section 2 describes the exact algorithm. Section 3 contains the computational results and Section 4 concludes the paper.

## 2. A Branch-and-Price for the CTOPTW

Section 2.1 provides a mathematical formulation of the CTOPTW. Section 2.2 describes the B&P scheme applied to the CTOPTW. Section 2.3 describes the implicit enumeration developed to solve pricing problems.

*2.1. Mathematical Formulation.* The CTOPTW arises in the directed graph $G = (P, A)$, where $P = \{0, 1, \ldots, n, n+1\}$ represents a set of points and $A = \{(i, j) \mid i, j \in P,\ i \neq j\}$ represents a set of arcs that connect pairs of points. $m$ identical players move from starting point 0 to ending point $n+1$. $P^r = \{1, \ldots, n\}$ represents the set of points where players can collect a reward $r_i$ by visiting a point $i \in P^r$. A player takes travelling time $t_{i,j}$ ($\geq 0$) to traverse an arc $(i, j) \in A$. The CTOPTW has the following constraints:

(C1) A point $i \in P^r$ can be visited only once by only one player.

(C2) A player cannot take more than time $L$ to move from 0 to $n+1$.

(C3) A point $i \in P^r$ has demand $d_i$ and a player can visit a set of points only if the sum of their demands does not exceed the player's capacity $Q$.

(C4) A player can visit $p_i$ only during the interval $[e_i, l_i]$. If the player arrives at $p_i$ before $e_i$, the player must wait until $e_i$. The player takes $u_i$ time to finish the visit at the point.

Let $H = \{h_1, h_2, \ldots\}$ be a set of paths from 0 to $n+1$ that satisfies (C1)–(C4). Using the mathematical formulation of [16, 17], the CTOPTW can be formulated in mathematical form as [CTOPTW].

$$\max \quad z = \sum_{h_i \in H} r(h_i)\, x_i, \tag{1}$$

$$\sum_{h_i \in H} a_{j,i} x_i \leq 1, \quad \forall j \in P^r, \tag{2}$$

$$\sum_{h_i \in H} x_i \leq m, \tag{3}$$

$$x_i \in \{0, 1\}, \ \forall h_i \in H, \tag{4}$$

where $r(h_i)$ represents the sum of collected rewards by traversing the path $h_i$; decision binary variable $x_i$ has the value 1 if the path $h_i$ is used and 0 otherwise; and $a_{j,i}$ is 1 if path $h_i$ visits point $p_j$ and 0 otherwise. The objective function (1) maximizes the sum of collected rewards. Constraint (2) ensures that no point is visited more than once. Constraint (3) limits the number of players to be less than or equal to $m$. Constraint (4) forces $x_i$ to be binary. This formulation has the same form as the set partitioning problem (SPP) with an exponential number of columns which is known to be NP-hard [36].

*2.2. Branch-and-Price.* Branch-and-price (B&P) is a special case of branch-and-bound (B&B) [37], which is an exact method used to find the optimal solution of an IP. The B&B transforms an IP to a linear problem (LP) by relaxing the integer constraints, solves the LP at the root node, and terminates if the optimal solution of the LP is an integer. Otherwise, the B&B branches the root node into some child nodes according to a predefined branching rule. For each child node, the B&B repeatedly solves an LP and decides whether it branches or bounds the child node. The repetition ends when the optimal solution of the IP is found. The B&B can be used to find the optimal solution of [CTOPTW]. If its LPs have an exponential number of columns, however, the B&B may fail.

B&P solves an LP at each node by the column generation technique. The column generation technique helps to find the optimal solution of an LP by considering only a small portion of its columns. In this paper, B&P is more appropriate than B&B because each of our benchmark instances has 102 points. B&P requires the user to define the branching rule. The simplest branching rule to fix a noninteger variable as 1 or 0 entails tremendous computational difficulty [16]. Therefore, the user may choose to define an alternative rule that forces a node or an arc to be used or not. Previous B&P papers about the TOP [16], the CTOP [30], and the TOPTW [17] all used the identical branching rule, and we also use it in this paper.

The column generation technique decomposes the original LP with many columns into master and pricing problems. The master problem contains only a small portion of the original columns. After the technique solves the master problem, it solves the pricing problem to detect a column that can improve the solution of the master problem. If it detects a column, it adds the column to the master problem as a new column. Otherwise, the technique terminates and concludes that the optimal solution of the master problem is also optimal for the original LP.

The pricing problem of the B&P in the CTOPTW is a form of the elementary shortest path problem with capacity and time windows (ESPPCTW). The ESPPCTW is a problem of determining the elementary path from the starting point to the ending point that collects the maximum net rewards while satisfying the two constraints. The ESPPCTW arises in the same graph with the CTOPTW, except that a point $i \in P^r$ has a dual reward $\bar{r}_i$ and its net reward is calculated as $r_i - \bar{r}_i$.

Let $\bar{r}(h_i)$ represent the sum of dual rewards that are collected by traversing path $h_i$. The optimal path $h^*$ of the ESPPCTW is found by solving

$$h^* = \arg\max_{h_i \in H} r(h_i) - \bar{r}(h_i). \tag{5}$$

*2.3. Implicit Enumeration.* The ESPPCTW is also NP-hard [38]. Full enumeration lists all feasible paths from 0 to $n+1$ and selects the path with the maximum net reward. In full enumeration, a state $s$ is defined as a path from 0 to $i \in P$ with a label $s = (i, r, q, t, E)$, where $i$ is the index of the last visited point, $r$ is the sum of rewards of the visited points, $q$ is the sum of demands of the visited points, $t$ is the current time of the path, and $E = (E_1, \ldots, E_n)$ is an $n$-sized binary vector in which $E_j = 1$ if the path has visited $j \in P^r$ and 0 otherwise.

$s$ can be extended to $j \in P$ if $q + d_j \le Q$, $t + t_{i,j} + u_i \le l_j$, and $E_j \ne 1$; when this is done, the label $s' = (j, r', q', t', E')$ of the extended state is updated as

$$\begin{aligned} r' &= r + r_j, \\ q' &= q + d_j, \\ t' &= \max\left(t + t_{i,j} + u_i, e_j\right), \\ E'_k &= \begin{cases} E_k & \text{if } k \ne j \\ 1 & \text{o.w.} \end{cases} \quad \text{for each } k \in P^r. \end{aligned} \tag{6}$$

Since full enumeration generates an exponential number of states, previous studies [39–42] have used a version of full enumeration called implicit enumeration to solve the ESPPCTW. Implicit enumeration allows domination between states. A state dominated by another state can be discarded without loss of optimality. As the number of dominated states increases, both the number of states generated and the computational effort decrease. Given two states $s = (i, r, q, t, E)$ and $s'' = (i, r'', q'', t'', E'')$, in the same node, $s$ dominates $s''$ if

$$\begin{aligned} r &\ge r'', \\ q &\le q'', \\ t &\le t'', \\ E_k &\le E''_k \quad \text{for each } p_k \in P^r. \end{aligned} \tag{7}$$

We use implicit enumeration to solve the pricing problem of the B&P or ESPPCTW but adopt two acceleration techniques to reduce the computational burden. The first is a heuristic algorithm proposed by Tae and Kim [17]. The aim of solving the ESPPCTW is to find an improving column. Although both implicit enumeration and the heuristic algorithm can find an improving column, each has a drawback. Implicit enumeration is computationally expensive but guarantees the optimality of the column, whereas the heuristic algorithm is computationally inexpensive but cannot guarantee optimality. Thus, we select implicit enumeration to solve the ESPPCTW when the heuristic algorithm fails to find an improving column.

The second technique is $ng$-route relaxation of Baldacci et al. [43], which is one type of elementary constraint relaxation. In this relaxation, each point $i$ has a group of points called $ng$-group or $ng_i$. $ng$-relaxed state $s_i$ in the point $i$ can be extended to a point $j \in P$ even though $s_i$ has visited $j$ if the extension is feasible and $j \notin ng_i$.

As a new state is generated in the implicit enumeration, the state is joined with $ng$-route relaxed states. This joining provides the upper bound of the state; if the upper bound is less promising than the lower bound, the state is ignored. $ng$-route relaxation has achieved great acceleration in previous papers [17, 42, 43]. $ng$-route relaxation requires that a constant $K$ be set. If $K$ is a large number, the relaxation provides strong bounds but requires extensive computational time to find them, whereas if $K$ is small, the relaxation provides weak bounds but requires less time [43]. Considering the tradeoffs, we set $K = 5$ [17].

## 3. Computational Results

Garcia et al. [31] made benchmark instances based on the instances of Solomon [44] and Cordeau et al. [45]. They tested their algorithm on these instances by setting the number of players to one and two. By the definition of the CTOPTW, the number of players should be at least two. However, they considered single player case because the case had been solved exactly by Righini and Salani [46]. In this paper, we only consider case of two players.

In each instance, each point has its own $x$-$y$ coordinates, demand, reward, and time windows. The travelling time between the points is calculated as Euclidean distance. The starting and ending points are located in the same positions in every instance. The constant of the time limitation constraint or $L$ is calculated as $l_{n+1} - e_0$ which is the end of the time window of the ending point minus the start of the time window of the starting point.

Table 1 compares the optimal solution of each instance with the best solutions reported in Garcia et al. [31] and Souffriau et al. [35]. We do not bring the solutions of Aghezzaf and Fahim [34] into this table since we have a reason to doubt that they used the different instances. Aghezzaf and Fahim [34] claimed they used the instances of Garcia et al. [31]. However, they considered $E_2$ as the capacity constraint while Garcia et al. [31] considered $E_1$ (here, we use the same notation with Garcia et al. [31]).

Notations in Table 1 represent the following: $n$, the number of players; $Q$, the value of vehicle capacity; TOPTW, the objective value found by TOPTW algorithms (see Garcia et al. [31]); ILS, the objective value found by Garcia et al.'s [31] iterated local search (ILS) algorithm; GRILS, the objective value found by Souffriau et al.'s [35] greedy randomized iterated local search (GRILS) algorithm; RUB, the value of upper bound at the root node found by our algorithm; UB, the value of upper bound found by our algorithm; LB, the value of lower bound found by our algorithm; CT, the value of computational time in seconds; "—" means no value is found.

Computational tests are performed using an Intel i7 3.6 GHz processor with 8 GB RAM. Our program is built on a Microsoft Visual Basic C++ environment and LPs are solved by IBM CPLEX. The program is run on a single core. Computation time is limited to 7200 seconds. All the benchmark instances and optimal solutions can be found in https://sites.google.com/site/optimizationlaboratory. The distance between points are calculated by using a Euclidean distance that is rounded to the first decimal for Solomon's instances (c101–c109, r101–r112, rc101–rc108) and to the second for Cordeau's instances (pr01–pr10).

We found all optimal solutions for Solomon's instances but only one for Coredeau. Coredeau's instances were more difficult to solve since they have wider time windows and more number of players. If the time windows are widened, the

Table 1: Computational results of Garcia et al.'s [31] instances with two players.

| Instance information | | | Previous algorithm | | | Proposed algorithm | | | |
|---|---|---|---|---|---|---|---|---|---|
| Name | $n$ | Q | TOPTW | ILS | GRILS | RUB | UB | LB | CT |
| c101 | 100 | 614 | 590 | 580 | 590 | 590 | 590 | 590 | 0.5 |
| c102 | 100 | 801 | 650 | 650 | 650 | 660 | 660 | 660 | 30.3 |
| c103 | 100 | 727 | 710 | 710 | 690 | 724.737 | 720 | 720 | 651.0 |
| c104 | 100 | 624 | 760 | 760 | 740 | 763.333 | 760 | 760 | 2494.4 |
| c105 | 100 | 656 | 640 | 640 | 640 | 640 | 640 | 640 | 2.0 |
| c106 | 100 | 604 | 620 | 620 | 620 | 620 | 620 | 620 | 12.9 |
| c107 | 100 | 648 | 670 | 660 | 670 | 670 | 670 | 670 | 17.1 |
| c108 | 100 | 641 | 680 | 680 | 680 | 680 | 680 | 680 | 23.9 |
| c109 | 100 | 788 | 710 | 710 | 710 | 720 | 720 | 720 | 37.4 |
| r101 | 100 | 380 | 341 | 322 | 344 | 344 | 344 | 344 | 0.2 |
| r102 | 100 | 684 | 501 | 508 | 504 | 508 | 508 | 508 | 57.2 |
| r103 | 100 | 566 | 513 | 512 | 515 | 518.5 | 517 | 517 | 2698.3 |
| r104 | 100 | 639 | 531 | 538 | 527 | 548 | 548 | 548 | 346.1 |
| r105 | 100 | 587 | 430 | 434 | 438 | 438 | 438 | 438 | 1.2 |
| r106 | 100 | 645 | 529 | 529 | 521 | 529 | 529 | 529 | 361.3 |
| r107 | 100 | 566 | 527 | 523 | 523 | 531 | 531 | 531 | 1657.2 |
| r108 | 100 | 639 | 534 | 539 | 538 | 555 | 555 | 555 | 4749.4 |
| r109 | 100 | 735 | 506 | 498 | 499 | 506 | 506 | 506 | 7.3 |
| r110 | 100 | 738 | 506 | 519 | 510 | 523 | 523 | 523 | 27.9 |
| r111 | 100 | 737 | 535 | 536 | 540 | 544 | 544 | 544 | 83.6 |
| r112 | 100 | 776 | 522 | 513 | 524 | 544 | 544 | 544 | 464.8 |
| rc101 | 100 | 557 | 421 | 427 | 427 | 427 | 427 | 427 | 0.3 |
| rc102 | 100 | 650 | 487 | 497 | 499 | 505 | 505 | 505 | 3.5 |
| rc103 | 100 | 527 | 512 | 501 | 510 | 520 | 520 | 520 | 6.3 |
| rc104 | 100 | 531 | 551 | 556 | 552 | 564 | 564 | 564 | 149.0 |
| rc105 | 100 | 592 | 451 | 448 | 469 | 480 | 480 | 480 | 1.0 |
| rc106 | 100 | 600 | 464 | 462 | 478 | 483 | 483 | 483 | 3.3 |
| rc107 | 100 | 531 | 520 | 516 | 506 | 526 | 526 | 526 | 23.2 |
| rc108 | 100 | 527 | 540 | 526 | 526 | 551 | 551 | 551 | 53.2 |
| pr01 | 48 | 506 | 481 | 489 | 496 | 502 | 502 | 502 | 77.5 |
| pr02 | 96 | 1001 | 685 | 654 | 672 | 0 | 0 | 0 | 7200.0 |
| pr03 | 144 | 1380 | 692 | 701 | 700 | 0 | 0 | 0 | 7200.0 |
| pr04 | 192 | 2523 | 880 | 872 | 836 | 0 | 0 | 0 | 7200.0 |
| pr05 | 240 | 3925 | 1031 | 1002 | 965 | 0 | 0 | 0 | 7200.0 |
| pr06 | 288 | 4198 | 992 | 952 | 888 | 0 | 0 | 0 | 7200.0 |
| pr07 | 72 | 585 | 560 | 547 | 557 | 0 | 0 | 0 | 7200.0 |
| pr08 | 144 | 1904 | 809 | 774 | 794 | 0 | 0 | 0 | 7200.0 |
| pr09 | 216 | 2967 | 819 | 828 | 795 | 0 | 0 | 0 | 7200.0 |
| pr10 | 288 | 4033 | 1037 | 998 | 955 | 0 | 0 | 0 | 7200.0 |

length of a feasible path (the number of the visited players by a path) increases. Then, the pricing problem or ESPPCTW becomes more hard to solve. Then, the B&P takes much more computational time to find the optimal solution. Similar computational results can be found in the paper of Tae and Kim [17] who solved the TOPTW by B&P. They also found many optimal solutions of Solomon's instances but only a few for Coredeau.

## 4. Conclusion

In this paper, we optimally solved some CTOPTW instances which has not been done previously. We showed that the B&P scheme of Boussier et al. [16], which was developed for the TOP, can be applied to the CTOPTW. To solve pricing problems, we used implicit enumeration acceleration techniques, heuristic algorithms [17], and *ng*-route relaxations [43]. We found 30 optimal solutions out of 39 benchmark instances of Garcia et al. [31].

## Competing Interests

The authors declare that they have no competing interests.

## Acknowledgments

This study has been funded by following two projects of KITECH (Korea Institute of Industrial Technology). Those projects are "Simulation Based Sewing Process Optimization Algorithm" and "A Development of the KLM System for the Stabilization of Kimchi Raw Material and Study on Food Industry."

## References

[1] B. L. Golden, L. Levy, and R. Vohra, "The orienteering problem," *Naval Research Logistics*, vol. 34, no. 3, pp. 307–318, 1987.

[2] P. Vansteenwegen, W. Souffriau, and D. Van Oudheusden, "The orienteering problem: a survey," *European Journal of Operational Research*, vol. 209, no. 1, pp. 1–10, 2011.

[3] I.-M. Chao, B. L. Golden, and E. A. Wasil, "The team orienteering problem," *European Journal of Operational Research*, vol. 88, no. 3, pp. 464–474, 1996.

[4] H. Tang and E. Miller-Hooks, "A TABU search heuristic for the team orienteering problem," *Computers and Operations Research*, vol. 32, no. 6, pp. 1379–1407, 2005.

[5] Z. Sevkli and F. E. Sevilgen, "Variable neighborhood search for the orienteering problem," in *Computer and Information Sciences—ISCIS 2006*, pp. 134–143, 2006.

[6] C. Archetti, A. Hertz, and M. G. Speranza, "Metaheuristics for the team orienteering problem," *Journal of Heuristics*, vol. 13, no. 1, pp. 49–76, 2007.

[7] L. Ke, C. Archetti, and Z. Feng, "Ants can solve the team orienteering problem," *Computers and Industrial Engineering*, vol. 54, no. 3, pp. 648–665, 2008.

[8] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden, "A guided local search metaheuristic for the team orienteering problem," *European Journal of Operational Research*, vol. 196, no. 1, pp. 118–127, 2009.

[9] H. Bouly, D.-C. Dang, and A. Moukrim, "A memetic algorithm for the team orienteering problem," *4OR*, vol. 8, no. 1, pp. 49–70, 2010.

[10] W. Souffriau, P. Vansteenwegen, G. Vanden Berghe, and D. Van Oudheusden, "A path relinking approach for the team orienteering problem," *Computers and Operations Research*, vol. 37, no. 11, pp. 1853–1859, 2010.

[11] D.-C. Dang, R. N. Guibadj, and A. Moukrim, "A pso-based memetic algorithm for the team orienteering problem," in *Applications of Evolutionary Computation*, pp. 471–480, Springer, 2011.

[12] S. Muthuswamy and S. S. Lam, "Discrete particle swarm optimization for the team orienteering problem," *Memetic Computing*, vol. 3, no. 4, pp. 287–303, 2011.

[13] B.-I. Kim, H. Li, and A. L. Johnson, "An augmented large neighborhood search method for solving the team orienteering problem," *Expert Systems with Applications*, vol. 40, no. 8, pp. 3065–3072, 2013.

[14] M. Poggi, H. Viana, and E. Uchoa, "The team orienteering problem: formulations and branch-cut and price," in *Proceedings of the 10th Workshop on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems (ATMOS '10)*, pp. 142–155, September 2010.

[15] D.-C. Dang, R. El-Hajj, and A. Moukrim, "A branch-and-cut algorithm for solving the team orienteering problem," in *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pp. 332–339, Springer, 2013.

[16] S. Boussier, D. Feillet, and M. Gendreau, "An exact algorithm for team orienteering problems," *4OR*, vol. 5, no. 3, pp. 211–230, 2007.

[17] H. Tae and B.-I. Kim, "A branch-and-price approach for the team orienteering problem with time windows," *International Journal of Industrial Engineering: Theory Applications and Practice*, vol. 22, no. 2, pp. 243–251, 2015.

[18] P. Vansteenwegen, W. Souffriau, G. Vanden Berghe, and D. Van Oudheusden, "Iterated local search for the team orienteering problem with time windows," *Computers & Operations Research*, vol. 36, no. 12, pp. 3281–3290, 2009.

[19] R. Montemanni and L. Gambardella, "An ant colony system for team orienteering problems with time windows," *Foundations of Computing and Decision Sciences*, vol. 34, pp. 287–306, 2009.

[20] N. Labadie, R. Mansini, J. Melechovský, and R. W. Calvo, "The team orienteering problem with time windows: an LP-based granular variable neighborhood search," *European Journal of Operational Research*, vol. 220, no. 1, pp. 15–27, 2012.

[21] S.-W. Lin and V. F. Yu, "A simulated annealing heuristic for the team orienteering problem with time windows," *European Journal of Operational Research*, vol. 217, no. 1, pp. 94–107, 2012.

[22] N. Labadie, J. Melechovský, and R. W. Calvo, "Hybridized evolutionary local search algorithm for the team orienteering problem with time windows," *Journal of Heuristics*, vol. 17, no. 6, pp. 729–753, 2011.

[23] Q. Hu and A. Lim, "An iterative three-component heuristic for the team orienteering problem with time windows," *European Journal of Operational Research*, vol. 232, no. 2, pp. 276–286, 2014.

[24] R. Montemanni, D. Weyland, and L. M. Gambardella, "An enhanced ant colony system for the team orienteering problem with time windows," in *Proceedings of the International Symposium on Computer Science and Society (ISCCS '11)*, pp. 381–384, IEEE, Kota Kinabalu, Malaysia, July 2011.

[25] N. Labadi, J. Melechovský, and R. Calvo, "An effective hybrid evolutionary local search for orienteering and team orienteering problems with time windows," in *Parallel Problem Solving from Nature, PPSN XI*, pp. 219–228, Springer, 2010.

[26] D. Gavalas, C. Konstantopoulos, K. Mastakas, G. Pantziou, and Y. Tasoulas, "Cluster-based heuristics for the team orienteering problem with time windows," in *Experimental Algorithms*, pp. 390–401, Springer, 2013.

[27] T. Cura, "An artificial bee colony algorithm approach for the team orienteering problem with time windows," *Computers & Industrial Engineering*, vol. 74, no. 1, pp. 270–290, 2014.

[28] C. D. Tarantilis, F. Stavropoulou, and P. P. Repoussis, "The capacitated team orienteering problem: a bi-level filter-and-fan method," *European Journal of Operational Research*, vol. 224, no. 1, pp. 65–78, 2013.

[29] Z. Luo, B. Cheang, A. Lim, and W. Zhu, "An adaptive ejection pool with toggle-rule diversification approach for the

capacitated team orienteering problem," *European Journal of Operational Research*, vol. 229, no. 3, pp. 673–682, 2013.

[30] C. Archetti, D. Feillet, A. Hertz, and M. G. Speranza, "The capacitated team orienteering and profitable tour problems," *Journal of the Operational Research Society*, vol. 60, no. 6, pp. 831–842, 2009.

[31] A. Garcia, P. Vansteenwegen, W. Souffriau, O. Arbelaitz, and M. Linaza, "Solving multi constrained team orienteering problems to generate tourist routes," Tech. Rep., Centre for Industrial Management, Katholieke Universiteit Leuven, Leuven, Belgium, 2009.

[32] B. Aghezzaf and H. El Fahim, "Solving the capacitated team orienteering problem with time windows through variable neighborhood search," *International Review on Computers and Software*, vol. 10, no. 11, pp. 1134–1142, 2015.

[33] Z. Li and X. Hu, "The orienteering problem with compulsory nodes and time window," in *Proceedings of the 8th International Conference on Service Systems and Service Management (ICSSSM '11)*, pp. 157–163, Tianjin, China, June 2011.

[34] B. Aghezzaf and H. El Fahim, "The multi-constraint team orienteering problem with time windows in the context of distribution problems: a variable neighborhood search algorithm," in *Proceedings of the 2nd IEEE International Conference on Logistics Operations Management (GOL '14)*, pp. 155–160, Rabat, Morocco, June 2014.

[35] W. Souffriau, P. Vansteenwegen, G. Vanden Berghe, and D. Van Oudheusden, "The multiconstraint team orienteering problem with multiple time windows," *Transportation Science*, vol. 47, no. 1, pp. 53–63, 2013.

[36] M. R. Garey and D. S. Johnson, *Computers and Intractability*, vol. 29, WH Freeman, New York, NY, USA, 2002.

[37] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*, Courier Corporation, 1982.

[38] M. Dror, "Note on the complexity of the shortest path models for column generation in VRPTW," *Operations Research*, vol. 42, no. 5, pp. 977–978, 1994.

[39] D. Feillet, P. Dejax, M. Gendreau, and C. Gueguen, "An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems," *Networks*, vol. 44, no. 3, pp. 216–229, 2004.

[40] G. Righini and M. Salani, "Symmetry helps: bounded bi-directional dynamic programming for the elementary shortest path problem with resource constraints," *Discrete Optimization*, vol. 3, no. 3, pp. 255–273, 2006.

[41] G. Righini and M. Salani, "Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming," *Computers & Operations Research*, vol. 36, no. 4, pp. 1191–1203, 2009.

[42] H. Tae and B.-I. Kim, "A multiple-starting-path approach to the resource-constrained k th elementary shortest path problem," *Mathematical Problems in Engineering*, vol. 2015, Article ID 803135, 7 pages, 2015.

[43] R. Baldacci, A. Mingozzi, and R. Roberti, "New route relaxation and pricing strategies for the vehicle routing problem," *Operations Research*, vol. 59, no. 5, pp. 1269–1283, 2011.

[44] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Operations Research*, vol. 35, no. 2, pp. 254–265, 1987.

[45] J.-F. Cordeau, M. Gendreau, and G. Laporte, "A tabu search heuristic for periodic and multi-depot vehicle routing problems," *Networks*, vol. 30, no. 2, pp. 105–119, 1997.

[46] G. Righini and M. Salani, "New dynamic programming algorithms for the resource constrained elementary shortest path problem," *Networks*, vol. 51, no. 3, pp. 155–170, 2008.