International Journal of Operational Research

ISSN online: 1745-7653 - ISSN print: 1745-7645
https://www.inderscience.com/ijor

Solving the team orienteering problem with time windows and mandatory visits using a constraint programming approach

Emre Kirac, Ridvan Gedik, Furkan Oztanriseven

DOI: 10.1504/IJOR.2020.10030954

Article History:

Received:	31 December 2019
Accepted:	20 February 2020
Published online:	26 January 2023

Solving the team orienteering problem with time windows and mandatory visits using a constraint programming approach

Emre Kirac

Luter School of Business,
Christopher Newport University,
One Avenue of the Arts, Newport News,
VA 23606, USA
Email: emre.kirac@cnu.edu

Ridvan Gedik*

Amazon.Com, Inc.,
410 Terry Ave N, Seattle 98109,
WA, USA
Email: rgedikn@gmail.com
*Corresponding author

Furkan Oztanriseven

Le Moyne College,
1419 Salt Springs Road, Syracuse,
NY 13214, USA
Email: oztanrf@lemoyne.edu

Abstract: This paper presents a constraint programming (CP) approach for solving the team orienteering problem with time windows and mandatory visits (TOPTW-MV), which has many real-world implementations, such as tourist tour planning, routing technicians, and disaster relief routing. In the TOPTW-MV, a set of locations is given; some locations must be visited, while others are optional. For each location, the profit, service time, and service time window information are known. A fleet of homogeneous vehicles is available for visiting locations and collecting the profits. The objective in solving this problem is to create a set of vehicle routes that begin and end at a depot, visit mandatory locations exactly once and optional locations at most once, while observing other restrictions such as time windows and sequence-based travel times. The CP-based approach finds 99 of the best-known solutions and explores 64 new best-known solutions for the benchmark instances.

Keywords: team orienteering problem; TOP; time windows; mandatory visits; vehicle routing; constraint programming.

Reference to this paper should be made as follows: Kirac, E., Gedik, R. and Oztanriseven, F. (2023) 'Solving the team orienteering problem with time windows and mandatory visits using a constraint programming approach', *Int. J. Operational Research*, Vol. 46, No. 1, pp.20–42.

Biographical notes: Emre Kirac is a Visiting Assistant Professor of Management at the Christopher Newport University. Prior to joining Christopher Newport University, he worked as an Assistant Professor of Engineering Management at the University Houston – Clear Lake. He received his PhD in Industrial Engineering from the University of Arkansas. He completed his Master's in Industrial and Systems Engineering from the Colorado State University-Pueblo. His research focuses on applying operations research tools and techniques to solve large-scale logistics planning problems arising in the humanitarian and supply chain sectors.

Ridvan Gedik is a Visiting Associate Professor in the Department of Business, Accounting and Sport Management at Elizabeth City State University. Previously, he worked as an Assistant Professor at the Quinnipiac University, University of New Haven and Mississippi State University. His primary research interests include developing efficient solution techniques for large-scale optimisation problems in supply chain management, logistics, healthcare, transportation, manufacturing and homeland security. He is also interested in developing machine learning models for transportation (road and airline) safety problems and sport games. Ridvan has several publications at different venues and secured funding from federal and state agencies for his research.

Furkan Oztanriseven is an Assistant Professor of Business Analytics at the Le Moyne College, NY. He completed his PhD in Industrial Engineering at the University of Arkansas. He received Master's in Business Administration and Industrial and Systems Engineering. His primary areas of research involve transportation and logistics, disruption response and homeland security, sustainability, and economic decision analysis. His research is funded by several notable governmental agencies such as the US Department of Homeland Security and the US Department of Transportation.

1 Introduction

The team orienteering problem with time windows and mandatory visits (TOPTW-MV) is a recently proposed version of the orienteering problem (OP), which originated from a sports game in which multiple players have to start at a control point, collect the rewards by visiting checkpoints and then return to the control point in a predetermined amount of time (Campos et al., 2014; Lin and Vincent, 2017; Tsiligirides, 1984). Players may or may not be able to visit all the nodes due to the time restriction; however, whoever collects the most rewards wins the game. Example of applications based on the OP and its extensions involve tourist routing problems (Souffriau et al., 2008; Souffriau and Vansteenwegen, 2010; Sylejmani et al., 2012), disaster relief logistics (Rath and Gutjahr, 2014; Kirac, 2016; Kirac and Milburn, 2018), pickup and delivery services (Gutiérrez-Jarpa et al., 2009; Ramirez-Marquez et al., 2010), and sales representative route planning (Tricoire et al., 2010). In this study, we will be using network terms; checkpoints and control points will be referred to as locations/nodes or visits, and depots, respectively.

In the TOPTW-MV, sets of mandatory and optional locations are given, each location with a profit, service time, and time window. A fixed maximum number of vehicle routes can be constructed, each with a given constant cost, to collect profit from these locations. Each optional location can be visited at most once, while all mandatory locations should be visited once within a given time window. Each vehicle route must start from and end at a depot. The time required to travel between locations is known. The objective of the TOPTW-MV is to find a set of routes that maximises the total profit.

The TOPTW-MV was first introduced by Lin and Vincent (2017). They constructed a mixed-integer linear programming (MIP) model and developed a multi-start simulated annealing (MSA) heuristic for solving the problem; the MSA heuristic has been updated from the one used for the TOP in Lin (2013). They also proposed 72 small problem instances to compare the MIP model and the MSA heuristic, and 168 large problem instances to compare the performance on the TOPTW of the MSA heuristic, the simulated annealing (SA) algorithm, and the artificial ant colony (ABC) algorithm proposed by Cura (2014). The MSA algorithm found better solutions than the ABC algorithm for 13 of the 72 small problem instances, while the remaining solutions were the same. For the large problem instances, the MSA heuristic arrived at 94 of the best-known solutions out of the 168 instances, while the SA and ABC algorithms arrived at 79 and 67 of the best-known solutions, respectively.

In this paper, we propose a new exact solution technique for the TOPTW-MV. We formulate the TOPTW-MV using a constraint programming (CP) model and refer to this model as the CP-TOPTW-MV model. CP has been shown to be an efficient solution technique for numerous combinatorial optimisation problems, such as parallel machine scheduling (Gökgür et al., 2018; Hooker, 2007; Jain and Grossmann, 2001; Nachtmann et al., 2014; Gedik et al., 2018), tournament organisation (Trick and Yildiz, 2011), kinematics problems (Pedamallu and Ozdamar, 2010), food processing and scheduling (Wari and Zhu, 2019), staff scheduling and rostering (Topaloglu and Ozkarahan, 2011; He and Qu, 2012), vehicle routing and traveling salesman problems (TSP) (Pesant et al., 1998, 1999; Quoc and Anh, 2010; Edirisinghe et al., 2010), and vehicle routing problem with time windows (VRPTW) (Shaw, 1998; De Backer et al., 2000; Rousseau et al., 2002, 2004; Guimarans et al., 2013). CP has also been successfully applied for solving the TOPTW (Gedik et al., 2017) and the multi-objective TOPTW (Hu et al., 2018).

The contributions of this paper are threefold. First, to the best of our knowledge, it introduces the first CP-based model for solving the TOPTW-MV. Due to the advantages of CP for expressing complex relationships, very difficult constraints such as mandatory and optional node visits, subtour elimination, and time windows are represented very efficiently. Compared with MILP formulations of the TOPTW-MV in Lin and Vincent (2017), the CP-TOPTW-MV model does not require a large number of decision variables and constraints. Thus, we are able to run benchmark instances without experiencing any memory problems. Also, unlike the sophisticated heuristic algorithms in the literature, the proposed CP model does not require extensive parameter tuning, as those methods do so. Second, the CP-TOPTW-MV model provides a promising base for other solution techniques for OP variants and related routing problems, potentially serving as a basis for new methodological developments. Third, we conducted computational experiments on the TOPTW-MV test instances in the literature and found that the CP-TOPTW-MV model outperforms current approaches in terms of solution quality. On average, it performs quite efficiently and obtains numerous best-known solutions. We found 99 of the best-known solutions out of 168 benchmark instances, and improved upon the best-known solution for 64 benchmark instances, finding a solution with an objective value strictly greater than the incumbent one. Furthermore, the proposed model proves that the best-known solution values for 15 benchmark test instances are indeed optimal.

The remainder of this paper is organised as follows. Section 2 reviews the latest literature regarding OPs. Section 3 formally presents the problem being addressed. Section 4 provides the CP formulation for the TOPTW-MV. Section 5 provides results for the CP-TOPTW-MV model and compares it to algorithms in the current literature. Finally, conclusions and future research directions are discussed in Section 6.

2 Literature review

The OP can be formulated as a special case of the well-known TSP (Matai et al., 2010; Jafari, 2019; Alipour and Razavi, 2019), but it differs from the OP, since the objective function minimises the traveling costs instead of maximising the total score collected. The team orienteering problem (TOP) is an extension of the OP; it presents the opportunity to collect profit by visiting the nodes with multiple vehicles and has also attracted researchers (Keshtkaran et al., 2016). The TOP is also a generalisation of the vehicle routing problem (VRP) where only a subset of nodes can be visited by a limited number of vehicles within a time budget (Sharma et al., 2018; Samadi-Dana et al., 2017). The orienteering problem with time windows (OPTW) and the team orienteering problem with time windows (TOPTW) extend the OP and TOP by incorporating time window constraints (Qian and Andrew, 2014). In the OPTW, every visit has to be performed within a given time frame (Duque et al., 2015; Kantor and Rosenwein, 1992). The TOPTW is a version of the OPTW, where profit can be collected with multiple vehicles to achieve the maximum gain (Cura, 2014; Gedik et al., 2017; Vansteenwegen et al., 2009). A comprehensive review of applications and solution techniques for the OP and its extensions are provided in Vansteenwegen et al. (2011) and Gunawan et al. (2016).

To provide the context for the problem, the latest literature on the OP and its variants is reviewed here. Gunawan et al. (2017) proposed two algorithms to solve the TOPTW, iterated local search (ILS) and a hybridisation of simulated annealing and iterated local search (SAILS). The authors demonstrated that ILS and SAILS improved the quality of the solutions and discovered 50 new best-known solutions. Park et al. (2017) presented a branch-and-price (B&P) algorithm for the capacitated team orienteering problem with time windows (CTOPTW). The authors show that the B&P scheme, which was originally developed for the TOP, can be applied to the CTOPTW. Verbeeck et al. (2017) introduced a time-dependent orienteering problem with time windows (TD-OPTW), where travel time is time-dependent and where each node has a time window and a service time. The authors proposed a mixed-integer problem formulation and an ant colony algorithm for the TD-OPTW. Vincent et al. (2017) proposed a new extension of the TOPTW called the multi-modal team orienteering problem with time windows (MM-TOPTW). The authors developed a particle swarm optimisation with multiple social learning terms to solve the MM-TOPTW. They demonstrated that the proposed algorithm produced optimal solutions for small and medium-scale instances and high-quality solutions for large-scale instances.

Pěnička et al. (2017) introduced the Dubins orienteering problem (DOP), which is a generalisation of the Euclidean OP. In the DOP, the reward-collecting tour has to accommodate the limited turning radius of the Dubins vehicle. Like the OP, the DOP maximises the total collected rewards by visiting a subset of the locations within a time budget. However, the DOP involves determining the vehicle's heading angle at each location to minimise the length of Dubins maneuvers between the target locations. Penicka et al. proposed a variable neighbourhood search (VNS), which was tested using existing OP benchmark instances. Bianchessi et al. (2018) presented a new compact formulation for the TOP and solved it by applying a branch-and-cut algorithm; the proposed algorithm was tested by comparing it with all the previously proposed exact methods. Archetti et al. (2018) studied the set orienteering problem (SOP), which is a generalisation of the OP, where nodes are grouped in clusters, and a score is associated with each cluster, which is collected only if at least one node from the cluster is visited. The authors proposed a metaheuristic composed of simple greedy and Tabu search algorithms. Pěnička et al. (2019) employed a VNS for the SOP. In addition, they proposed an integer linear programming formulation of the SOP to find the optimal solution for small and medium-sized problems. Santini (2019) applied the adaptive large neighbourhood search algorithm to solve the OP. The results showed that the implemented heuristic provided high-quality solutions on long runs, compared to other state-of-the-art heuristics. Conversely, the exact algorithms found a higher number of best-known solutions than the proposed heuristic on the short run (with a time limit of five hours). Gavalas et al. (2019) proposed two cluster-based ILS algorithms for the tourist trip design problem (TTDP), which is a common variant of the TOPTW. In this approach, customer locations were grouped based on geographical areas, thereby increasing the probability of visiting the customers. The authors claimed that their approach improved the solution quality and execution time for the TTDP instances. Vincent et al. (2019) studied the team orienteering problem with time windows and time-dependent scores (TOPTW-TDS), which is an extension of the TOPTW. The problem maximises the total collected scores based on the time-dependent scores for different periods of the visited points of interest. They proposed a hybrid algorithm (HABC) to solve the TOPTW-TDS by hybridising the artificial bee colony algorithm with a SA-based acceptance rule. None of these studies considers a set of mandatory, with some optional, nodes.

To the best of our knowledge, Lin and Vincent (2017) is the only published work on the TOPTW-MV in the literature. However, some studies of other variants of the TOPTW-MV have been reported. Tricoire et al. (2010) modelled the multi-period OP with multiple time windows to plan the routes of sales representatives, involving both mandatory and optional customers. Mandatory customers were long-term customers that were visited on a regular basis, while optional customers were potential new customers. Each customer had several time windows, which could change from day to day. A VNS algorithm was developed to solve the problem. Salazar-Aguilar et al. (2014) studied the multi-district team orienteering problem (MDTOP), which was introduced to support road maintenance activities. In this problem, the maintenance region is divided into districts, and activities are classified as either mandatory or optional within each district. The problem was formulated as a mixed-integer linear problem and solved with an adaptive large neighbourhood search algorithm. Palomo-Martínez et al. (2017b) inspired the work of Salazar-Aguilar et al. (2014); they proposed the orienteering problem with mandatory visits and exclusionary constraints (OPMVEC), which involves mandatory visits and optional nodes. A hybrid algorithm based on a reactive GRASP and a general VNS were developed to solve the problem. Palomo-Martínez et al. (2017a) adapted five mixed-integer linear formulations from the TSP literature to solve the the OP with mandatory visits and conflicts (OPMVC), which handled the subtour elimination constraints were handled differently. These five different formulations for the OPMVC were compared and tested. Most recently, Lu et al. (2018) proposed a memetic algorithm (MA) for the OPMVEC. The computational results demonstrate that in terms of the number of best-known solutions found, the MA outperformed both the VNS algorithm and the exact solution methods in the literature.

3 Problem statement

The formal definition of the TOPTW-MV can be given as follows. Let's assume that we have a directed graph denoted as $G = (\mathcal{N}, \mathcal{A})$, in which $\mathcal{N} = \{0, 1, ..., n\}$ stands for the set of nodes (customers) and $\mathcal{A} = \{(i, j) : i \in \mathcal{N}, j \in \mathcal{N}, i \neq j\}$ for the set of arcs connecting nodes. Among customers, set K represents the set of important customers that must be visited, such that $K \subset N$. The remaining set of customers $(N \setminus K)$ can be visited selectively. The depot is represented as the node $i = 0$. The time it takes to travel between node pairs i and j is denoted as t_{ij}. Furthermore, p_i and s_i stand for the profit collected and the service time spent, respectively, upon visiting customer $i \in \mathcal{N}$. The time interval $[b_i, e_i]$ for each node i describes the time window during which a visit can be performed. Thus, b_i and e_i are the earliest and latest times when a visit can be performed, respectively. The number of vehicles is not fixed, and therefore is a variable defined as δ. We assign a sufficiently large integer δ' as an upper limit on this variable. We note that all vehicles are identical. The TOPTW-MV seeks to identify a set of vehicle tours (H) in which important customers are visited and other customers are visited only once at most. Each tour $h \in H$ should start at the depot after b_0 and end before e_0. The fixed cost of using a vehicle is γ, and the objective in solving the TOPTW-MV is to maximise the total net profit, which can be calculated by subtracting the $\delta \times \gamma$ from the total profit collected from the customers. An extra node $n + 1$ (dummy depot) is created and added to set $\mathcal{N}$, in order to model the return of the vehicles to the depot. Travel times are created, as in $t_{i,n+1} = t_{i,0}$, for all $i \in \mathcal{N}$, and the time windows for visits are set by $e_{n+1} = e_0$ and $b_{n+1} = b_0$. Furthermore, there is no profit associated with visiting the depot, and it is assumed that no time is spent at the depot ($p_0 = p_{n+1} = s_0 = s_{n+1} = 0$).

4 Methodology

Lin and Yu (2012) demonstrated that solving the TOPTW in polynomial time is unlikely, since it is proven to be an NP-hard problem. Hence, solving the TOPTW-MV in polynomial time is not possible, since it can be easily reduced to the TOPTW. For this reason, only approximate solution approaches have been developed to tackle realistic size instances of the TOPTW-MV (see Lin and Vincent, 2017).

Pure CP modelling (Gedik et al., 2017) and hybrid CP and mathematical programming applications (Gedik et al., 2016) have been very successful at efficiently solving combinatorial scheduling and routing problems. Focacci et al. (2002) pointed out that CP models perform more efficiently compared to their mathematical model counterparts on combinatorial problems when more side constraints are added to the problem. More specifically, a traditional branch-and-cut approach is much more successful at solving instances of the TSP than the pure CP model. However, CP models have become more competitive at solving the TSP with time windows. Gedik et al. (2017) reported that the same behaviour exists for tackling the TOPTW with pure CP models. Their CP model dominated both pure mathematical model formulations and other state-of-the-art heuristics. This section describes our efforts to develop a pure CP model formulation to solve the TOPTW-MV.

The mathematical formulation of the TOPTW-MV can be seen in Lin and Vincent (2017), and therefore, it is not given in this paper. *interval variable*, which is a CP-based decision variable that represents the start, end, duration time, and resource usage of a scheduling activity or task (Laborie et al., 2018), is utilised to produce the necessary CP model formulation. The interval variable can also carry binary existence information for activities, which is a key feature that enables us to model whether or not an activity is scheduled/visited on/by a specific resource/vehicle. More specifically, an interval variable can be made *optional*, and therefore would be *present* in the solution and *absent* if not (IBM, 2014). Since the binary feature is already embedded in the definition of the interval variable via optionality, no extra effort is necessary to model it, as in traditional mathematical programming (Laborie and Rogerie, 2008). An interval variable's final status can be easily retrieved by using *presenceOf(Interval Variable)*, which will return true/1 or false/0, depending on the application domain.

We defined and used the following list of interval variables in the CP formulation.

- For each vehicle $h \in H$, we define u_h as an *optional interval variable*.
- For each $i \in \mathcal{N} \setminus K \cup \{i = 0, i = n+1\}$, y_i is defined as an *optional interval variable*. For other nodes $\{i | i \in K \cup \{i = 0, i = n+1\}\}$, y_i is defined as a *mandatory interval variable* (i.e., it must be present in the solution). Note that the corresponding service time s_i is automatecially assigned to y_i.
- For each customer $\{i \in \mathcal{N} | i \neq 0, i \neq n+1\}$, we define x_{ih} as an *optional interval variable* that represents the possibility of visiting customer i with vehicle h and spending a duration of service time of s_i.
- $Z_i = \{x_{i1}, x_{i2}, ..., x_{ih}, ..., x_{i\delta}\}$ represents the combination of all vehicles in terms of interval variables that can be used to visit node $\{i \in \mathcal{N} | i \neq 0, i \neq n+1\}$. Note that δ is a decision variable.
- $Q_h = \{x_{1h}, x_{2h}, ..., x_{ih}, ..., x_{nh}\}$ stands for the customers $\{i \in \mathcal{N} | i \neq 0, i \neq n+1\}$ that can possibly be visited by vehicle $h \in H$. This structure is also referred to as the *interval sequence variable*.

The domains of the interval variables are the times when activities may start, end, or lapse. A feasible solution is obtained in a CP model when the domains of all of the decision variables, including the interval variables, are filtered (propagated) into a single value. Unlike traditional mathematical programming, CP is much more expressive and can model complex restrictions via its *global constraints*. A global constraint is a collection of simple constraints, such as $=$, $\leq$, $\geq$, and logical functions. Each global constraint has powerful domain-filtering algorithms that run simultaneously on the set of restrictions defined in the global constraint to reduce the domains of each associated decision variable to a single value (Vilím, 2007). However, there is no standard way to define a global constraint across all CP platforms. Thus, a CP model generated via IBM's CP Optimizer may not be generated in the exact same way in other CP solvers such as Gecode. This study utilises IBM's CP Optimizer and its global constraints to formulate and solve the TOPTW-MV, since there has been extensive research conducted to solve a variety of combinatorial problems on this platform (see Laborie et al., 2018 for several applications). More specifically, we used *alternative*, *cumulative*, *NoOverlap*, and *span* global constraints in the pure CP model (CP-TOPTW-MV) to capture the restrictions of the TOPTW-MV. We refer the reader to Laborie and Rogerie (2008) and Laborie and Rogerie (2008) for the mathematical descriptions and other potential usages of these global constraints. The resulting CP model is as follows:

$$\text{maximise} \quad \sum_{i=1}^{i=n} p_i y_i - \delta \times \gamma$$

subject to (CP-TOPTW-MV)

$$\delta = \sum_{h=1}^{h=\delta'} presenceOf(u_h) \tag{1}$$

$$Alternative(y_i, Z_i) \qquad i \in \mathcal{N}, i \neq 0, i \neq n+1 \tag{2}$$

$$Cumulative(\{y_0, y_1, ..., y_{n+1}\}, 1, \delta) \tag{3}$$

$$y_i.StartMin = b_i \qquad i \in \mathcal{N} \tag{4}$$

$$y_i.StartMax = e_i \qquad i \in \mathcal{N} \tag{5}$$

$$NoOverlap(Q_h, (t_{ij} | i \in N, j \in N)) \quad h \in H \tag{6}$$

$$Q_h.First(y_0) \qquad h \in H \tag{7}$$

$$Q_h.Last(y_{n+1}) \qquad h \in H \tag{8}$$

$$Span(u_h, Q_h) \qquad h \in H \tag{9}$$

The objective of the CP-TOPTW-MV model is to maximise the total net profit. The number of vehicles used is determined in constraint (1). Recall that u_h is defined as an optional interval variable, and that there is one for each vehicle h. If a vehicle is used, its corresponding u_h variable will be in the final solution and thus $presenceOf(u_h)$ will return one. The sum of these queries across all vehicles is made equal to δ in constraint (1). Constraint (2) ensures that each important customer is visited by exactly one vehicle and that other regular customers are visited by at most one vehicle. This relationship can be represented by one *alternative* global constraint, as indicated in constraint (2), due to the definitions of the interval variables. Note that y_i for each

$i \in K$ is defined as a mandatory interval variable and therefore must be present in the solution. Consequently, one of the vehicles represented as an interval variable in Z_i must be assigned to visit customer $i \in K$. For all of the other customers ($i \in \mathcal{N} \setminus K \cup \{i = 0, i = n + 1\}$), this constraint reserves only one vehicle from the set Z_i if y_i to be present in the solution. Ensuring that the total number of busy vehicles may not exceed δ is handled by the *cumulative* global constraint (3). Constraints (4) and (5) assign the earliest and latest visit start times, respectively, for each customer in $i \in \mathcal{N}$. The *NoOverlap* global constraint ensures that the necessary travel time (t_{ij}) is used to visit customer j after visiting customer i. Constraint (6) also ensures that the interval variables in set Q_h (the set of customer visits) do not overlap with each other. This guarantees the sequential visiting of the customers. Constraints (7) and (8) make the depot the first and the last visit for each vehicle, respectively. Finally, the *span* global constraint is utilised to monitor whether a vehicle, h, is ever used to visit any customer in Q_h. By definition, the interval represented by the optional interval variable u_h spans all present intervals from the set Q_h. Since there is no service time assigned to u_h, it will start and end with the first and last visits in Q_h. The interval variable u_h, representing vehicle h, will be absent (not used) only when all the interval variables in Q_h are absent from the solution.

As one might realise, the CP-TOPTW-MV formulation is very compact; it includes a low number of constraints, due to the nature of the interval variable concept. One interval variable may carry information for the start, end, duration, and resource consumption rate of an activity and provide an opportunity to mimic binary behaviour, unlike continuous or integer decision variables in traditional mathematical programming. In this way, we were able to load and solve even the largest instances of the TOPTW-MV with no memory problems.

5 Computational results

This section presents the computational study and its results. We first describe benchmark instances of the TOPTW-MV, and then present and discuss the computational results.

5.1 Experiments

For this study, the performance of the CP-TOPTW-MV model was compared with that of the MSA, the SA, and the ACO algorithms discussed by Lin and Vincent (2017) on the TOPTW-MV test instances from Lin and Vincent (2017), which were generated based on the OPTW instances designed by Righini and Salani (2009). Among these instances, 56 large problem instances were converted from ‘the Solomon instances’ (Solomon, 1987) (sets c*_100, r*_100, rc*_100, c*_200, r*_200 and rc*_200), as they are referred to in the literature. These 56 large problem instances were run for a path cost of 75, 100, and 125. Therefore, a total of 168 large problem instances were used. All of these TOPTW-MV instances had 100 nodes and 5 mandatory visits.

The time it took to visit each location pair was rounded down to the first decimal for the TOPTW-MV instances, as in Lin and Vincent (2017). Recall that the interval variables defined in Section 4 have the domain values of the visit start, end, or lapse time. Fractional distance values may therefore initiate domain ranges with an infinite range, and therefore, they need to be treated in the CP context. Hence, every rounded-down travel time value was later multiplied by 10 to limit the domains of the interval variables to the set of integer values. We also scaled up the other time-related parameters, such as service time and time window start and end times (i.e., each value was multiplied by 10). This re-scaling helped the propagation algorithms to start filtering the domains of the interval variables and move towards a feasible solution, at the cost of searching in an enlarged domain range.

Table 1 Comparison of CP-TOPTW-MV to state-of-the-art algorithms on TOPTW-MV instances

Instance set	*#INS*	*CP-TOPTW-MV*		*MSA*		*SA*		*ACO*	
(5 mandatory visits)		*#BKS*	*Gap (%)*	*#BKS*	*Gap (%)*	*#BKS*	*Gap (%)*	*#BKS*	*Gap (%)*
γ = *75*									
c100	9	5	0.48	5	0.59	3	1.03	8	0.12
r100	12	0	3.24	4	0.50	4	0.94	4	0.56
rc100	8	2	3.33	5	0.41	3	0.68	4	0.44
c200	8	8	0.00	0	4.10	0	4.26	0	5.32
r200	11	9	0.81	5	1.85	8	2.10	1	6.21
rc200	8	5	1.05	1	2.75	2	3.17	0	4.18
γ = *100*									
c100	9	5	0.81	2	1.25	3	1.11	8	0.13
r100	12	3	3.02	3	1.47	3	1.69	4	0.95
rc100	8	3	4.33	1	1.27	3	0.87	4	1.06
c200	8	8	0.00	0	3.97	0	4.06	0	7.45
r200	11	10	0.13	6	1.70	7	2.63	1	9.17
rc200	8	5	1.75	4	1.03	1	3.67	1	5.26
γ = *125*									
c100	9	2	1.63	2	1.41	2	1.41	9	0.00
r100	12	8	1.29	2	3.02	2	3.02	3	2.74
rc100	8	4	4.55	4	2.07	4	2.07	0	2.65
c200	8	8	0.00	0	3.66	0	3.66	0	8.71
r200	11	10	0.05	4	4.08	4	4.08	2	10.83
rc200	8	4	3.11	6	0.12	6	0.12	3	2.49
All	168	99	1.64	54	1.96	55	2.25	52	3.79

Table 2 Optimal solution values for Solomon TOPTW-MV instances proved by CP-TOPTW-MV

Instance name	γ	*# path*	*Optimal solution*	*Time (s)*
c201	75	3	1,585	3.1
c202	75	3	1,585	15.6
c204	75	3	1,585	30.0
c205	75	3	1,585	3.7
c206	75	3	1,585	13.1
c208	75	3	1,585	4.0
r204	75	2	1,308	2,925.3
c201	100	3	1,510	3.3
c205	100	3	1,510	3.9
c206	100	3	1,510	14.3
c208	100	3	1,510	4.0
c201	125	3	1,435	3.7
c202	125	3	1,435	16.6
c205	125	3	1,435	4.7
c208	125	3	1,435	4.3

The CP-TOPTW-MV model was implemented in C++, using IBM ILOG CP Optimizer 12.6. All of the experiments were conducted on an Intel Core i5 equipped with 2.70 GHz and 8 GB of RAM. The CP-TOPTW-MV model provides the optimal solution if it can be found in less than 30 minutes. If an optimal solution is not identified in 30 minutes then the CP-TOPTW model ends the search with the best feasible solution identified within the time limit.

5.2 Results

To ensure fair comparisons, the solutions for each algorithm were compared with the best-known solutions (BKS) among MSA, SA, ACO, and CP-TOPTW-MV. While for these approaches, the average objective value over 5 replicates is presented, as reported in Lin and Vincent (2017), the CP does not require replications, so the results reported represent only a single objective value for each test instance.

Table 1 summarises the results for the TOPTW-MV instances for path costs (γ) of 75, 100, and 125. The #INS column gives the number of instances in a set. The gap (%) column shows the average percentage gap between solutions produced by the algorithm and the best-known solutions obtained across all the algorithms. The number of best-known solutions found by each algorithm is given in the #BKS column. We can see from Table 1 that, on average, the CP-TOPTW-MV model offers the best performance on this metric, with an average gap of of 1.64%, while MSA, SA, and ACO provide average gaps of 1.96%, 2.25%, and 3.79%, respectively. The CP-TOPTW-MV model obtained best-known solutions for 99 out of 168 instances, whereas MSA, SA, and ACO could found the best-known solutions for only 54, 55, and 53 instances, respectively. Overall, the CP-TOPTW-MV model outperforms the other three approaches in terms of the number of best-known solutions found and the average gap from the best-known solutions. When γ is increased, the CP-TOPTW-MV model outperforms other algorithms even more in terms of the number of best-known solutions. For example, when γ is 125, the number of best-known solutions obtained by the CP-TOPTW-MV model (36) is double the number of best-known solutions obtained by both the MSA heuristic (18) and the SA algorithm (18).

Table 3 Paired t-test

	Problem set	*Difference*	*StDev*	*Dof*	*p-value*
CP vs. MSA	γ = 75	8.75	44.02	55	0.071
	γ = 100	6.66	41.9	55	0.120
	γ = 125	9.54	48.76	55	0.075
	Overall	8.32	44.73	167	0.009
CP vs. SA	γ = 75	12.34	47.93	55	0.030
	γ = 100	14.54	47.67	55	0.013
	γ = 125	17.34	57.19	55	0.014
	Overall	14.74	50.86	167	0.000
CP vs. ACO	γ = 75	24.48	52.09	55	0.000
	γ = 100	37.98	67.26	55	0.000
	γ = 125	38.29	85.5	55	0.001
	Overall	33.58	69.51	167	0.000

Table 4 Detailed results for Solomon instances with five mandatory visits and γ = 75

Name	*BKS*	*CP-TOPTW-MV*				*MSA*				*SA*				*ACO*			
		Best	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*
c101	815	805	1.23	51	5	805	1.23	50	5	815	0.00	51	5	815	0.00	51	5
c102	*955*	*955*	0.00	58	5	945	1.05	56	5	935	2.09	55	5	945	1.05	56	5
c103	995	985	1.01	58	5	995	0.00	56	5	985	1.01	56	5	995	0.00	57	5
c104	1,025	1,025	0.00	60	5	1,005	1.95	57	5	1,015	0.98	58	5	1,025	0.00	59	5
c105	855	855	0.00	52	5	855	0.00	52	5	845	1.17	51	5	855	0.00	52	5
c106	875	865	1.14	54	5	875	0.00	52	5	875	0.00	52	5	875	0.00	52	5
c107	905	905	0.00	55	5	905	0.00	54	5	905	0.00	54	5	905	0.00	54	5
c108	925	925	0.00	57	5	915	1.08	55	5	915	1.08	55	5	925	0.00	56	5
c109	1,015	1,005	0.99	58	5	1,015	0.00	59	5	985	2.96	56	5	1,015	0.00	59	5
r101	310	302	2.58	35	5	310	0.00	36	5	308	0.65	34	5	304	1.94	34	5
r102	566	551	2.65	49	5	560	1.06	49	5	557	1.59	48	5	566	0.00	51	5
r103	657	654	0.46	54	5	654	0.46	52	5	651	0.91	53	5	657	0.00	54	5
r104	737	709	3.80	57	5	728	1.22	57	5	709	3.80	57	5	737	0.00	56	5
r105	511	510	0.20	44	5	504	1.37	43	5	506	0.98	43	5	511	0.00	45	5
r106	655	616	5.95	52	5	655	0.00	51	5	643	1.83	50	5	649	0.92	50	5
r107	670	648	3.28	52	5	670	0.00	54	5	667	0.45	52	5	669	0.15	54	5
r108	736	706	4.08	57	5	736	0.00	56	5	728	1.09	55	5	728	1.09	55	5
r109	621	595	4.19	50	5	616	0.81	49	5	621	0.00	50	5	617	0.64	50	5
r110	651	636	2.30	52	5	650	0.15	51	5	651	0.00	52	5	650	0.15	51	5
r111	684	669	2.19	53	5	680	0.58	56	5	684	0.00	53	5	676	1.17	54	5
r112	748	694	7.22	56	5	745	0.40	56	5	748	0.00	56	5	743	0.67	55	5
rc101	583	583	0.00	45	5	583	0.00	45	5	583	0.00	45	5	583	0.00	45	5
rc102	693	666	3.90	48	5	693	0.00	50	5	689	0.58	51	5	683	1.44	48	5
rc103	755	721	4.50	47	5	755	0.00	51	5	740	1.99	49	5	755	0.00	50	5
rc104	851	797	6.35	58	5	846	0.59	54	5	851	0.00	53	5	847	0.47	53	5
rc105	656	638	2.74	48	5	644	1.83	45	5	651	0.76	48	5	656	0.00	48	5
rc106	666	666	0.00	52	5	666	0.00	50	5	666	0.00	50	5	661	0.75	48	5
rc107	730	710	2.74	51	5	730	0.00	49	5	724	0.82	49	5	724	0.82	50	5
rc108	839	785	6.44	54	5	832	0.83	52	5	828	1.31	51	5	839	0.00	53	5

Table 4 Detailed results for Solomon instances with five mandatory visits and γ = 75 (continued)

Name	*BKS*	*CP-TOPTW-MV*				*MSA*				*SA*				*ACO*			
		Best	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*
c201	*1,585*	*1,585*	0.00	100	3	1,510	4.73	100	4	1,515	4.42	93	3	1,510	4.73	100	4
c202	*1,585*	*1,585*	0.00	100	3	1,525	3.79	94	3	1,510	4.73	100	4	1,510	4.73	100	4
c203	*1,585*	*1,585*	0.00	100	3	1,510	4.73	100	4	1,510	4.73	100	4	1,510	4.73	100	4
c204	*1,585*	*1,585*	0.00	100	3	1,510	4.73	100	4	1,510	4.73	100	4	1,510	4.73	100	4
c205	*1,585*	*1,585*	0.00	100	3	1,535	3.15	95	3	1,535	3.15	95	3	1,510	4.73	100	4
c206	*1,585*	*1,585*	0.00	100	3	1,515	4.42	93	3	1,525	3.79	94	3	1,510	4.73	100	4
c207	*1,585*	*1,585*	0.00	100	3	1,545	2.52	96	3	1,510	4.73	100	4	1,435	9.46	100	5
c208	*1,585*	*1,585*	0.00	100	3	1,510	4.73	100	4	1,525	3.79	94	3	1,510	4.73	100	4
r201	1,198	1,158	3.34	100	4	1,183	1.25	93	3	1,198	0.00	93	3	1,158	3.34	100	4
r202	1,226	1,158	5.55	100	4	1,226	0.00	97	3	1,226	0.00	97	3	1,158	5.55	100	4
r203	1,233	1,233	0.00	100	3	1,233	0.00	100	3	1,233	0.00	100	3	1,158	6.08	100	4
r204	*1,308*	*1,308*	0.00	100	2	1,158	11.47	100	4	1,158	11.47	100	4	1,158	11.47	100	4
r205	1,233	1,233	0.00	100	3	1,233	0.00	100	3	1,233	0.00	100	3	1,158	6.08	100	4
r206	1,233	1,233	0.00	100	3	1,227	0.49	97	3	1,233	0.00	100	3	1,158	6.08	100	4
r207	1,233	1,233	0.00	100	3	1,230	0.24	98	3	1,233	0.00	100	3	1,233	0.00	100	3
r208	*1,308*	*1,308*	0.00	100	2	1,233	5.73	100	3	1,158	11.47	100	4	1,158	11.47	100	4
r209	1,233	1,233	0.00	100	3	1,218	1.22	94	3	1,233	0.00	100	3	1,158	6.08	100	4
r210	1,233	1,233	0.00	100	3	1,233	0.00	100	3	1,231	0.16	99	3	1,158	6.08	100	4
r211	1,233	1,233	0.00	100	3	1,233	0.00	100	3	1,233	0.00	100	3	1,158	6.08	100	4
rc201	1,448	1,424	1.66	100	4	1,434	0.97	92	3	1,448	0.00	93	3	1,424	1.66	100	4
rc202	1,489	1,424	4.37	100	4	1,489	0.00	99	3	1,484	0.34	98	3	1,424	4.37	100	4
rc203	*1,499*	*1,499*	0.00	100	3	1,424	5.00	100	4	1,424	5.00	100	4	1,424	5.00	100	4
rc204	*1,499*	*1,499*	0.00	100	3	1,424	5.00	100	4	1,424	5.00	100	4	1,424	5.00	100	4
rc205	1,459	1,424	2.40	100	4	1,449	0.69	92	3	1,459	0.00	94	3	1,424	2.40	100	4
rc206	*1,499*	*1,499*	0.00	100	3	1,494	0.33	98	3	1,424	5.00	100	4	1,424	5.00	100	4
rc207	*1,499*	*1,499*	0.00	100	3	1,424	5.00	100	4	1,424	5.00	100	4	1,424	5.00	100	4
rc208	*1,499*	*1,499*	0.00	100	3	1,424	5.00	100	4	1,424	5.00	100	4	1,424	5.00	100	4
Avg			1.56				1.60				1.94				2.78		

Table 5 Detailed results for Solomon instances with five mandatory visits and $\gamma = 100$

Name	*BKS*	*CP-TOPTW-MV*				*MSA*				*SA*				*ACO*			
		Best	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*
c101	690	680	1.45	51	5	680	1.45	50	5	670	2.90	50	5	690	0.00	51	5
c102	*830*	*830*	0.00	58	5	810	2.41	56	5	820	1.20	56	5	820	1.20	56	5
c103	870	850	2.30	58	5	860	1.15	56	5	860	1.15	56	5	870	0.00	57	5
c104	900	900	0.00	60	5	890	1.11	58	5	890	1.11	58	5	900	0.00	59	5
c105	730	730	0.00	52	5	710	2.74	50	5	720	1.37	50	5	730	0.00	52	5
c106	750	740	1.33	53	5	750	0.00	52	5	750	0.00	53	5	750	0.00	52	5
c107	780	780	0.00	54	5	780	0.00	54	5	780	0.00	54	5	780	0.00	54	5
c108	790	790	0.00	56	5	780	1.27	54	5	790	0.00	55	5	790	0.00	55	5
c109	890	870	2.25	57	5	880	1.12	58	5	870	2.25	57	5	890	0.00	59	5
r101	182	176	3.30	35	5	173	4.95	34	5	182	0.00	34	5	182	0.00	35	5
r102	*442*	*442*	0.00	50	5	425	3.85	48	5	422	4.52	48	5	434	1.81	50	5
r103	526	514	2.28	54	5	526	0.00	53	5	522	0.76	51	5	522	0.76	52	5
r104	601	589	2.00	56	5	593	1.33	56	5	594	1.16	56	5	601	0.00	56	5
r105	*386*	*386*	0.00	45	5	376	2.59	44	5	375	2.85	43	5	374	3.11	43	5
r106	518	483	6.76	52	5	518	0.00	49	5	509	1.74	49	5	514	0.77	49	5
r107	548	525	4.20	53	5	544	0.73	52	5	540	1.46	54	5	548	0.00	53	5
r108	603	580	3.81	58	5	592	1.82	54	5	589	2.32	55	5	603	0.00	57	5
r109	494	438	11.34	50	5	492	0.40	50	5	494	0.00	50	5	491	0.61	50	5
r110	*527*	*527*	0.00	54	5	521	1.14	51	5	515	2.28	50	5	517	1.90	52	5
r111	571	565	1.05	57	5	571	0.00	55	5	553	3.15	54	5	560	1.93	54	5
r112	613	604	1.47	55	5	608	0.82	54	5	613	0.00	55	5	610	0.49	56	5
rc101	458	458	0.00	45	5	458	0.00	45	5	458	0.00	45	5	458	0.00	45	5
rc102	*565*	*565*	0.00	50	5	549	2.83	46	5	563	0.35	48	5	557	1.42	47	5
rc103	616	606	1.62	50	5	614	0.32	50	5	616	0.00	49	5	612	0.65	49	5
rc104	727	681	6.33	59	5	726	0.14	53	5	726	0.14	53	5	727	0.00	54	5
rc105	526	514	2.28	49	5	520	1.14	46	5	526	0.00	47	5	519	1.33	48	5
rc106	*555*	*555*	0.00	52	5	546	1.62	50	5	549	1.08	51	5	527	5.05	47	5
rc107	610	513	15.90	51	5	596	2.30	49	5	596	2.30	49	5	610	0.00	50	5
rc108	716	655	8.52	56	5	703	1.82	53	5	694	3.07	53	5	716	0.00	53	5

Table 5 Detailed results for Solomon instances with five mandatory visits and $\gamma = 100$ (continued)

Name	*BKS*	*CP-TOPTW-MV*				*MSA*				*SA*				*ACO*			
		Best	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*
c201	*1,510*	*1,510*	0.00	100	3	1,470	2.65	96	3	1,460	3.31	95	3	1,410	6.62	100	4
c202	*1,510*	*1,510*	0.00	100	3	1,450	3.97	94	3	1,440	4.64	93	3	1,410	6.62	100	4
c203	*1,510*	*1,510*	0.00	100	3	1,410	6.62	100	4	1,430	5.30	92	3	1,310	13.25	100	5
c204	*1,510*	*1,510*	0.00	100	3	1,410	6.62	100	4	1,410	6.62	100	4	1,410	6.62	100	4
c205	*1,510*	*1,510*	0.00	100	3	1,480	1.99	97	3	1,470	2.65	96	3	1,410	6.62	100	4
c206	*1,510*	*1,510*	0.00	100	3	1,460	3.31	95	3	1,470	2.65	96	3	1,410	6.62	100	4
c207	*1,510*	*1,510*	0.00	100	3	1,430	5.30	92	3	1,460	3.31	95	3	1,410	6.62	100	4
c208	*1,510*	*1,510*	0.00	100	3	1,490	1.32	98	3	1,450	3.97	94	3	1,410	6.62	100	4
r201	*1,141*	*1,141*	0.00	97	3	1,138	0.26	97	3	1,119	1.93	95	3	1,058	7.27	100	4
r202	1,158	1,158	0.00	100	3	1,152	0.52	98	3	1,158	0.00	100	3	1,058	8.64	100	4
r203	1,158	1,158	0.00	100	3	1,158	0.00	100	3	1,158	0.00	100	3	1,058	8.64	100	4
r204	*1,258*	*1,258*	0.00	100	2	1,151	8.51	99	3	1,155	8.19	98	3	1,058	15.90	100	4
r205	1,158	1,158	0.00	100	3	1,158	0.00	100	3	1,158	0.00	100	3	1,058	8.64	100	4
r206	1,175	1,158	1.45	100	3	1,158	1.45	100	3	1,175	0.00	81	2	1,058	9.96	100	4
r207	1,158	1,158	0.00	100	3	1,158	0.00	100	3	1,158	0.00	100	3	1,158	0.00	100	3
r208	*1,258*	*1,258*	0.00	100	2	1,158	7.95	100	3	1,058	15.90	100	4	1,058	15.90	100	4
r209	1,158	1,158	0.00	100	3	1,158	0.00	100	3	1,158	0.00	100	3	1,058	8.64	100	4
r210	1,158	1,158	0.00	100	3	1,158	0.00	100	3	1,158	0.00	100	3	1,058	8.64	100	4
r211	1,158	1,158	0.00	100	3	1,158	0.00	100	3	1,124	2.94	92	3	1,058	8.64	100	4
rc201	1,368	1,324	3.22	100	4	1,368	0.00	92	3	1,355	0.95	90	3	1,324	3.22	100	4
rc202	1,414	1,324	6.36	100	4	1,414	0.00	99	3	1,324	6.36	100	4	1,324	6.36	100	4
rc203	1,424	1,424	0.00	100	3	1,324	7.02	100	4	1,324	7.02	100	4	1,424	0.00	100	3
rc204	1,424	1,424	0.00	100	3	1,424	0.00	100	3	1,324	7.02	100	4	1,324	7.02	100	4
rc205	1,385	1,324	4.40	100	4	1,385	0.00	95	3	1,384	0.07	93	3	1,324	4.40	100	4
rc206	*1,424*	*1,424*	0.00	100	3	1,421	0.21	99	3	1,411	0.91	96	3	1,324	7.02	100	4
rc207	*1,424*	*1,424*	0.00	100	3	1,412	0.84	97	3	1,324	7.02	100	4	1,324	7.02	100	4
rc208	1,424	1,424	0.00	100	3	1,422	0.14	99	3	1,424	0.00	100	3	1,324	7.02	100	4
Avg			1.67				1.75				2.28				3.99		

Table 6 Detailed results for Solomon instances with five mandatory visits and γ = 125

Name	*BKS*	*CP-TOPTW-MV*				*MSA*				*SA*				*ACO*			
		Best	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*
c101	565	555	1.77	51	5	565	0.00	51	5	565	0.00	51	5	565	0.00	51	5
c102	705	685	2.84	56	5	695	1.42	57	5	695	1.42	57	5	705	0.00	58	5
c103	745	725	2.68	58	5	735	1.34	56	5	735	1.34	56	5	745	0.00	57	5
c104	775	775	0.00	60	5	765	1.29	58	5	765	1.29	58	5	775	0.00	59	5
c105	605	605	0.00	52	5	595	1.65	50	5	595	1.65	50	5	605	0.00	52	5
c106	625	615	1.60	53	5	615	1.60	52	5	615	1.60	52	5	625	0.00	52	5
c107	655	645	1.53	56	5	655	0.00	54	5	655	0.00	54	5	655	0.00	54	5
c108	675	655	2.96	55	5	665	1.48	55	5	665	1.48	55	5	675	0.00	56	5
c109	765	755	1.31	58	5	735	3.92	57	5	735	3.92	57	5	765	0.00	59	5
r101	*70*	*70*	0.00	36	5	60	14.29	28	4	60	14.29	28	4	60	14.29	28	4
r102	*317*	*317*	0.00	50	5	310	2.21	49	5	310	2.21	49	5	309	2.52	49	5
r103	*400*	*400*	0.00	54	5	373	6.75	51	5	373	6.75	51	5	392	2.00	51	5
r104	*469*	*469*	0.00	57	5	468	0.21	55	5	468	0.21	55	5	463	1.28	55	5
r105	*251*	*251*	0.00	45	5	250	0.40	45	5	250	0.40	45	5	245	2.39	43	5
r106	*393*	*393*	0.00	50	5	376	4.33	48	5	376	4.33	48	5	377	4.07	49	5
r107	425	402	5.41	54	5	425	0.00	53	5	425	0.00	53	5	415	2.35	53	5
r108	477	459	3.77	58	5	474	0.63	56	5	474	0.63	56	5	477	0.00	56	5
r109	*374*	*374*	0.00	52	5	371	0.80	50	5	371	0.80	50	5	365	2.41	51	5
r110	392	383	2.30	53	5	392	0.00	50	5	392	0.00	50	5	392	0.00	53	5
r111	*440*	*440*	0.00	57	5	419	4.77	54	5	419	4.77	54	5	433	1.59	53	5
r112	498	478	4.02	58	5	489	1.81	56	5	489	1.81	56	5	498	0.00	56	5
rc101	*333*	*333*	0.00	45	5	313	6.01	43	5	313	6.01	43	5	330	0.90	45	5
rc102	*449*	*449*	0.00	51	5	439	2.23	49	5	439	2.23	49	5	436	2.90	48	5
rc103	503	477	5.17	52	5	503	0.00	50	5	503	0.00	50	5	501	0.40	50	5
rc104	605	547	9.59	57	5	605	0.00	53	5	605	0.00	53	5	589	2.64	55	5
rc105	403	398	1.24	49	5	403	0.00	48	5	403	0.00	48	5	373	7.44	44	5
rc106	*430*	*430*	0.00	52	5	415	3.49	48	5	415	3.49	48	5	409	4.88	47	5
rc107	461	367	20.39	50	5	461	0.00	47	5	461	0.00	47	5	454	1.52	48	5
rc108	*581*	*581*	0.00	54	5	553	4.82	53	5	553	4.82	53	5	578	0.52	52	5

Table 6 Detailed results for Solomon instances with five mandatory visits and γ = 125 (continued)

Name	*BKS*	*CP-TOPTW-MV*				*MSA*				*SA*				*ACO*			
		Best	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*	*Best*	*Gap (%)*	*#visited*	*#path*
c201	*1,435*	*1,435*	0.00	100	3	1,375	4.18	94	3	1,375	4.18	94	3	1,310	8.71	100	4
c202	*1,435*	*1,435*	0.00	100	3	1,395	2.79	96	3	1,395	2.79	96	3	1,310	8.71	100	4
c203	*1,435*	*1,435*	0.00	100	3	1,345	6.27	91	3	1,345	6.27	91	3	1,310	8.71	100	4
c204	*1,435*	*1,435*	0.00	100	3	1,355	5.57	92	3	1,355	5.57	92	3	1,310	8.71	100	4
c205	*1,435*	*1,435*	0.00	100	3	1,395	2.79	96	3	1,395	2.79	96	3	1,310	8.71	100	4
c206	*1,435*	*1,435*	0.00	100	3	1,395	2.79	96	3	1,395	2.79	96	3	1,310	8.71	100	4
c207	*1,435*	*1,435*	0.00	100	3	1,385	3.48	95	3	1,385	3.48	95	3	1,310	8.71	100	4
c208	*1,435*	*1,435*	0.00	100	3	1,415	1.39	98	3	1,415	1.39	98	3	1,310	8.71	100	4
r201	*1,066*	*1,066*	0.00	97	3	1,052	1.31	95	3	1,052	1.31	95	3	1,025	3.85	92	3
r202	*1,083*	*1,083*	0.00	100	3	1,077	0.55	98	3	1,077	0.55	98	3	1,077	0.55	98	3
r203	1,083	1,083	0.00	100	3	1,083	0.00	100	3	1,083	0.00	100	3	1,083	0.00	100	3
r204	*1,208*	*1,208*	0.00	100	2	1,083	10.35	100	3	1,083	10.35	100	3	958	20.70	100	4
r205	1,089	1,083	0.55	100	3	1,089	0.00	82	2	1,089	0.00	82	2	958	12.03	100	4
r206	*1,184*	*1,184*	0.00	95	2	1,161	1.94	90	2	1,161	1.94	90	2	958	19.09	100	4
r207	*1,206*	*1,206*	0.00	99	2	1,083	10.20	100	3	1,083	10.20	100	3	1,083	10.20	100	3
r208	*1,208*	*1,208*	0.00	100	2	1,083	10.35	100	3	1,083	10.35	100	3	958	20.70	100	4
r209	1,083	1,083	0.00	100	3	1,083	0.00	100	3	1,083	0.00	100	3	1,083	0.00	100	3
r210	1,083	1,083	0.00	100	3	1,083	0.00	100	3	1,083	0.00	100	3	958	11.54	100	4
r211	*1,205*	*1,205*	0.00	99	2	1,083	10.12	100	3	1,083	10.12	100	3	958	20.50	100	4
rc201	1,303	1,224	6.06	100	4	1,303	0.00	94	3	1,303	0.00	94	3	1,301	0.15	94	3
rc202	1,335	1,224	8.31	100	4	1,335	0.00	98	3	1,335	0.00	98	3	1,328	0.52	97	3
rc203	1,349	1,349	0.00	100	3	1,349	0.00	100	3	1,349	0.00	100	3	1,349	0.00	100	3
rc204	1,349	1,349	0.00	100	3	1,349	0.00	100	3	1,349	0.00	100	3	1,224	9.27	100	4
rc205	*1,334*	*1,334*	0.00	98	3	1,329	0.37	96	3	1,329	0.37	96	3	1,324	0.75	97	3
rc206	*1,349*	*1,349*	0.00	100	3	1,341	0.59	98	3	1,341	0.59	98	3	1,224	9.27	100	4
rc207	1,424	1,349	5.27	100	3	1,424	0.00	100	4	1,424	0.00	100	4	1,424	0.00	100	4
rc208	1,424	1,349	5.27	100	3	1,424	0.00	100	4	1,424	0.00	100	4	1,424	0.00	100	4
Avg			1.64				2.51				2.51				4.69		

There are 15 instances for which the CP-TOPTW-MV model was able to prove the optimality of the solution obtained. The optimalities of these instances were previously unreported in the literature. Table 2 provides the optimal solution value, along with the number of paths and the time required by the CP-TOPTW-MV model to return the solution and prove its optimality for each instance. The average computation time of the CP-TOPTW-MV model for the set of instances reported in Table 2 is 203.3 seconds. Without the instance of r204 when γ = 75, the average computation time is even less, only 8.9 seconds. Another primary advantage of the CP-TOPTW-MV model over any other heuristic algorithm is that it states whether the obtained solution is optimal, whereas the heuristic algorithms cannot.

Detailed results for all of the algorithms for the TOPTW-MV instances are given in Tables 4, 5, and 6 for the cases when γ was 75, 100, and 125, respectively. The first column provides the instance name, and the second gives the best-known solution value, BKS. The second (best) and third [gap (%)] columns provide the objective function value and percent gap to BKS for each instance, respectively. The third column (#visited) illustrates the number of jobs visited in a given problem instance. Finally, the fourth column (#path) presents the number of paths generated for the best solution identified. Bolded text indicates the new best-known solutions found in Tables 4, 5, and 6. The CP-TOPTW-MV model discovered 64 new best solutions. Judging from these results, the model usually achieves a better average percentage gap than MSA, SA, and ACO for all of these instances. The gap percentage performance of the CP-TOPTW-MV model for a cost of path of 125 is better than for a cost of path of 100, and only 6% worse than for a path cost of 75. The complexity of the model increases to enable it to find the optimal solution with an increase in the cost of path; the CP-TOPTW-MV model is the only model that can handle this increase in complexity.

In order to analyse whether the performance difference between the CP-TOPTW-MV model and the MSA heuristic, the SA algorithm, and the ACO algorithms is significant, we performed one-sided paired t-tests with a 95% confidence interval. The results are illustrated in Table 3 for cost-of-path scenarios of 75, 100, 125, and overall. These statistical analyses were conducted using the best objective function values found by the algorithms. The difference column shows that the difference between the CP-TOPTW-MV model and the other approaches is always positive, meaning that the average objective function value calculated for the problem instances by the CP-TOPTW-MV model is always greater in comparison to the others. The statistical results of the paired t-test, in the p-value column, show that, at a confidence level of 95%, there is a statically significant mean difference between the CP-TOPTW-MV model and the SA and ACO algorithms (p-value < 0.05). This indicates that the CP-TOPTW-MV model statistically outperforms the SA and ACO algorithms in terms of maximising overall profit. While there is no significant mean difference between the CP-TOPTW-MV model and the MSA heuristic when γ = 75, 100, and 125 individually, for the overall results, the performance of the CP-TOPTW-MV model is statistically better (p-value < 0.05) than that of the MSA heuristic. In summary, the proposed CP-TOPTW-MV statistically produces results superior to those of MSA, SA, and ACO for the benchmark instances.

The proposed CP-based model outperforms other approaches in the literature in solving a very practical problem, the TOPTW-MV. Depending on the context, businesses can use our model to boost their profitability by using our model. This will lead to a competitive advantage over their competitors. An increase in profitability can be transferred to the shareholders as well as customers by reducing costs due to improved profit margins. Moreover, governments may end up receiving more corporate income taxes due to higher net incomes, as well as an increase in tax revenues from increased dividends. Furthermore, planning customer visits with a more efficiently use of the CP-TOPTW-MV model can help reduce emissions. Another important managerial implication of our results is that companies, like Amazon, need to deliver products daily, and a consideration of time windows may take advantage of the running speed of the CP-TOPTW-MV model to run day-to-day operations more efficiently by taking into account potentially rapid changes in time windows. Furthermore, the CP-TOPTW-MV model could be much more productive in disaster relief, because being able to run an optimisation model much faster and more efficiently will save more lives, possibly hundreds, depending on the nature of the disaster. Our results clearly showed that adopting the CP-TOPTW-MV model as a solution method will magnify benefits (e.g., increasing profits or saving lives) even further, which makes it eminently applicable to complex real-life situations.

6 Conclusions

In this paper, we examine the TOPTW-MV, develop a CP model and compare the results of this model with state-of-the-art algorithms using benchmark instances. The new model, the CP-TOPTW-MV model, employs only a few constraints by taking advantage of interval variables. By carrying multiple information on interval variables, such as start times, end times, durations, and usage rates, the CP-TOPTW-MV model is able to solve instances of complex routing in the TOPTW-MV. The computational results indicate that the model outperforms the current approaches in the literature in terms of average performance and the number of identified best-known solutions. Our CP-TOPTW-MV model found 64 new best solutions. Moreover, the proposed model proved that 15 of the best-known solutions are optimal. The new results we report herein can serve as benchmarks for future studies. The framework is robust and can accommodate additional problem components. If the components were to be adapted to accept new constraints, the framework would also be applicable to solving other variants of the OP.

This study possesses some limitations and can be extended in several future directions. We observed that the CP-TOPTW-MV model fails to identify, within a given time limit, whether a solution is just a feasible solution or an optimal solution. This is due to the initiation of several possible domain-filtering attempts during the search for a better solution. We believe that the CP-TOPTW-MV model's performance can be improved in a hybrid setting where more challenging feasibility-related constraints are handled by CP, and the objective function or the constraints related to solution quality are handled via a bounding technique such as linear relaxation. A logic-based Benders decomposition approach may very well improve the quality of the solutions if the feasibility sub-problems are solved efficiently. Similarly, an adaptive heuristic method may very well be used to guide the search towards better objective function values, while feasibility constraints are handled through the use of a CP-based approach.

References

Alipour, M.M. and Razavi, S.N. (2019) 'A new local search heuristic based on nearest insertion into the convex hull for solving euclidean TSP', *International Journal of Operational Research*, Vol. 34, No. 3, pp.409–429.

Archetti, C., Carrabs, F. and Cerulli, R. (2018) 'The set orienteering problem', *European Journal of Operational Research*, Vol. 267, No. 1, pp.264–272.

Bianchessi, N., Mansini, R. and Speranza, M.G. (2018) 'A branch-and-cut algorithm for the team orienteering problem', *International Transactions in Operational Research*, Vol. 25, No. 2, pp.627–635.

Campos, V., Martí, R., Sánchez-Oro, J. and Duarte, A. (2014) 'Grasp with path relinking for the orienteering problem', *The Journal of the Operational Research Society*, Vol. 65, No. 12, pp.1800–1813 [online] http://www.jstor.org/stable/24505185.

Cura, T. (2014) 'An artificial bee colony algorithm approach for the team orienteering problem with time windows', *Computers & Industrial Engineering*, Vol. 74, No. 1, pp.270–290.

De Backer, B., Furnon, V., Shaw, P., Kilby, P. and Prosser, P. (2000) 'Solving vehicle routing problems using constraint programming and metaheuristics', *Journal of Heuristics*, Vol. 6, No. 4, pp.501–523.

Duque, D., Lozano, L. and Medaglia, A.L. (2015) 'Solving the orienteering problem with time windows via the pulse framework', *Computers & Operations Research*, Vol. 54, No. 1, pp.168–176.

Edirisinghe, N.C.P., Bowers, M.R. and Agarwal, A. (2010) 'Generalised pickup and delivery problem with dynamic time windows', *International Journal of Operational Research*, Vol. 8, No. 1, pp.85–109 [online] https://www.inderscienceonline.com/doi/abs/10.1504/IJOR.2010.033105.

Focacci, F., Lodi, A. and Milano, M. (2002) 'Mathematical programming techniques in constraint programming: a short overview', *Journal of Heuristics*, Vol. 8, No. 1, pp.7–17.

Gavalas, D., Konstantopoulos, C., Mastakas, K., Pantziou, G.E. et al. (2019) 'Efficient cluster-based heuristics for the team orienteering problem with time windows', *APJOR*, Vol. 36, No. 1, p.1950001.

Gedik, R., Rainwater, C., Nachtmann, H. and Pohl, E.A. (2016) 'Analysis of a parallel machine scheduling problem with sequence dependent setup times and job availability intervals', *European Journal of Operational Research*, Vol. 251, No. 2, pp.640–650 [online] http://www.sciencedirect.com/science/article/pii/S0377221715010693.

Gedik, R., Kirac, E., Milburn, A.B. and Rainwater, C. (2017) 'A constraint programming approach for the team orienteering problem with time windows', *Computers & Industrial Engineering*, Vol. 107, pp.178–195 [online] http://www.sciencedirect.com/science/article/pii/S0360835217301134.

Gedik, R., Kalathia, D., Egilmez, G. and Kirac, E. (2018) 'A constraint programming approach for solving unrelated parallel machine scheduling problem', *Computers & Industrial Engineering*, Vol. 121, No. 1, pp.139–149.

Gökgür, B., Hnich, B. and Özpeynirci, S. (2018) 'Parallel machine scheduling with tool loading: a constraint programming approach', *International Journal of Production Research*, Vol. 56, No. 16, pp.5541–5557 [online] https://doi.org/10.1080/00207543.2017.1421781.

Guimarans, D., Herrero, R., Ramos, J. and Padrón, S. (2013) 'Solving vehicle routing problems using constraint programming and lagrangean relaxation in a metaheuristics framework', *Management Innovations for Intelligent Supply Chains*, pp.123–143.

Gunawan, A., Lau, H.C. and Vansteenwegen, P. (2016) 'Orienteering problem: a survey of recent variants, solution approaches and applications', *European Journal of Operational Research*, Vol. 255, No. 2, pp.315–332.

Gunawan, A., Lau, H.C., Vansteenwegen, P. and Lu, K. (2017) 'Well-tuned algorithms for the team orienteering problem with time windows', *Journal of the Operational Research Society*, Vol. 68, No. 8, pp.861–876.

Gutiérrez-Jarpa, G., Marianov, V. and Obreque, C. (2009) 'A single vehicle routing problem with fixed delivery and optional collections', *IIE Transactions*, Vol. 41, No. 12, pp.1067–1079.

He, F. and Qu, R. (2012) 'A constraint programming based column generation approach to nurse rostering problems', *Computers & Operations Research*, Vol. 39, No. 12, pp.3331–3343.

Hooker, J.N. (2007) 'Good and bad futures for constraint programming (and operations research)', *Constraint Programming Letters*, Vol. 1, No. 1, pp.21–32.

Hu, W., Fathi, M. and Pardalos, P.M. (2018) 'A multi-objective evolutionary algorithm based on decomposition and constraint programming for the multi-objective team orienteering problem with time windows', *Applied Soft Computing*, Vol. 73, No. 1, pp.383–393.

IBM (2014) *IBM ILOG CPLEX Optimization Studio V12.6* [online] http://www-01.ibm.com/support/knowledgecenter/SSSA5P/welcome (accessed 15 November 2017).

Jafari, N. (2019) 'A non-assignment problem-based formulation for the asymmetric travelling salesman problem and its variation', *International Journal of Operational Research*, Vol. 35, No. 4, pp.575–585.

Jain, V. and Grossmann, I.E. (2001) 'Algorithms for hybrid milp/cp models for a class of optimization problems', *INFORMS Journal on Computing*, Vol. 13, No. 4, pp.258–276.

Kantor, M.G. and Rosenwein, M.B. (1992) 'The orienteering problem with time windows', *Journal of the Operational Research Society*, Vol. 43, No. 6, pp.629–635.

Keshtkaran, M., Ziarati, K., Bettinelli, A. and Vigo, D. (2016) 'Enhanced exact solution methods for the team orienteering problem', *International Journal of Production Research*, Vol. 54, No. 2, pp.591–601 [online] https://doi.org/10.1080/00207543.2015.1058982.

Kirac, E. and Milburn, A.B. (2018) 'A general framework for assessing the value of social data for disaster response logistics planning', *European Journal of Operational Research*, Vol. 269, No. 2, pp.486–500.

Kirac, E. (2016) *Incorporating A New Class of Uncertainty in Disaster Relief Logistics Planning*, PhD thesis, University of Arkansas.

Laborie, P. and Rogerie, J. (2008) 'Reasoning with conditional time-intervals', in *FLAIRS Conference*, pp.555–560.

Laborie, P., Rogerie, J., Shaw, P. and Vilím, P. (2018) 'IBM ILOG CP optimizer for scheduling', *Constraints*, Vol. 23, No. 2, pp.210–250.

Lin, S-W. (2013) 'Solving the team orienteering problem using effective multi-start simulated annealing', *Applied Soft Computing*, Vol. 13, No. 2, pp.1064–1073.

Lin, S-W. and Vincent, F.Y. (2017) 'Solving the team orienteering problem with time windows and mandatory visits by multi-start simulated annealing', *Computers & Industrial Engineering*, Vol. 114, No. 1, pp.195–205.

Lin, S-W. and Yu, V.F. (2012) 'A simulated annealing heuristic for the team orienteering problem with time windows', *European Journal of Operational Research*, Vol. 217, No. 1, pp.94–107.

Lu, Y., Benlic, U. and Wu, Q. (2018) 'A memetic algorithm for the orienteering problem with mandatory visits and exclusionary constraints', *European Journal of Operational Research*, Vol. 268, No. 1, pp.54–69.

Matai, R., Singh, S.P. and Mittal, M.L. (2010) 'Traveling salesman problem: an overview of applications, formulations, and solution approaches', *Traveling Salesman Problem, Theory and Applications*, Vol. 1, No. 1, pp.1–25.

Nachtmann, H., Mitchell, K.N., Rainwater, C., Gedik, R. and Pohl, E.A. (2014) 'Optimal dredge fleet scheduling within environmental work windows', *Transportation Research Record: Journal of the Transportation Research Board*, Vol. 2426, No. 1, pp.11–19.

Pěnička, R., Faigl, J. and Saska, M. (2019) 'Variable neighborhood search for the set orienteering problem and its application to other orienteering problem variants', *European Journal of Operational Research*, Vol. 276, No. 3, pp.816–825.

Pěnička, R., Faigl, J., Váňa, P. and Saska, M. (2017) 'Dubins orienteering problem', *IEEE Robotics and Automation Letters*, Vol. 2, No. 2, pp.1210–1217.

Palomo-Martínez, P.J., Salazar-Aguilar, M.A. and Albornoz, V.M. (2017a) 'Formulations for the orienteering problem with additional constraints', *Annals of Operations Research*, Vol. 258, No. 2, pp.503–545.

Palomo-Martínez, P.J., Salazar-Aguilar, M.A., Laporte, G. and Langevin, A. (2017b) 'A hybrid variable neighborhood search for the orienteering problem with mandatory visits and exclusionary constraints', *Computers & Operations Research*, Vol. 78, No. 1, pp.408–419.

Park, J., Lee, J., Ahn, S., Bae, J. and Tae, H. (2017) 'Exact algorithm for the capacitated team orienteering problem with time windows', *Mathematical Problems in Engineering*, Vol. 2017, No. 1, pp.1–6.

Pedamallu, C.S. and Ozdamar, L. (2010) 'A symbolic-interval inference approach for constraint satisfaction: implementation on kinematics applications', *International Journal of Operational Research*, Vol. 8, No. 2, pp.127–149.

Pesant, G., Gendreau, M., Potvin, J-Y. and Rousseau, J-M. (1998) 'An exact constraint logic programming algorithm for the traveling salesman problem with time windows', *Transportation Science*, Vol. 32, No. 1, pp.12–29.

Pesant, G., Gendreau, M., Potvin, J-Y. and Rousseau, J-M. (1999) 'On the flexibility of constraint programming models: from single to multiple time windows for the traveling salesman problem', *European Journal of Operational Research*, Vol. 117, No. 2, pp.253–263.

Qian, H. and Andrew, L. (2014) '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.

Quoc, N.H.B. and Anh, D.T. (2010) 'Combining constraint programming and tabu search to solve vehicle routing problem with time windows', in *2010 IEEE RIVF International Conference on Computing and Communication Technologies, Research, Innovation, and Vision for the Future (RIVF)*, pp.1–4.

Ramirez-Marquez, J.E., Kulturel-Konak, S. and Sanseverino, C.M.R. (2010) 'Probabilistic solution discovery algorithm for the orienteering problem', *International Journal of Industrial and Systems Engineering*, Vol. 6, No. 1, pp.45–61 [online] https://www.inderscienceonline.com/doi/abs/10.1504/IJISE.2010.033996.

Rath, S. and Gutjahr, W.J. (2014) 'A math-heuristic for the warehouse location – routing problem in disaster relief', *Computers & Operations Research*, Vol. 42, No. 1, pp.25–39.

Righini, G. and Salani, M. (2009) 'Decremental state space relaxation strategies and initialization heuristics for solving the orienteering problem with time windows with dynamic programming', *Computers & Operations Research*, Vol. 4, No. 4, pp.1191–1203.

Rousseau, L-M., Gendreau, M. and Pesant, G. (2002) 'Using constraint-based operators to solve the vehicle routing problem with time windows', *Journal of Heuristics*, Vol. 8, No. 1, pp.43–58.

Rousseau, L-M., Gendreau, M., Pesant, G. and Focacci, F. (2004) 'Solving vrptws with constraint programming based column generation', *Annals of Operations Research*, Vol. 130, Nos. 1–4, pp.199–216.

Salazar-Aguilar, M.A., Langevin, A. and Laporte, G. (2014) 'The multi-district team orienteering problem', *Computers & Operations Research*, Vol. 41, No. 1, pp.76–82.

Samadi-Dana, S., Paydar, M.M., Jouzdani, J. et al. (2017) 'A simulated annealing solution method for robust school bus routing', *International Journal of Operational Research*, Vol. 28, No. 3, pp.307–326.

Santini, A. (2019) 'An adaptive large neighbourhood search algorithm for the orienteering problem', *Expert Systems with Applications*, Vol. 123, No. 1, pp.154–167.

Sharma, S.K., Routroy, S. and Yadav, U. (2018) 'Vehicle routing problem: recent literature review of its variants', *International Journal of Operational Research*, Vol. 33, No. 1, pp.1–31.

Shaw, P. (1998) 'Using constraint programming and local search methods to solve vehicle routing problems', in *International Conference on Principles and Practice of Constraint Programming*, Springer, pp.417–431.

Solomon, M.M. (1987) 'Algorithms for the vehicle routing and scheduling problems with time window constraints', *Operations Research*, Vol. 35, No. 2, pp.254–265.

Souffriau, W. and Vansteenwegen, P. (2010) 'Tourist trip planning functionalities: state-of-the-art and future', in *International Conference on Web Engineering*, Springer, pp.474–485.

Souffriau, W., Vansteenwegen, P., Vertommen, J., Berghe, G.V. and Oudheusden, D.V. (2008) 'A personalized tourist trip design algorithm for mobile tourist guides', *Applied Artificial Intelligence*, Vol. 22, No. 10, pp.964–985.

Sylejmani, K., Dorn, J. and Musliu, N. (2012) 'A Tabu search approach for multi constrained team orienteering problem and its application in touristic trip planning', in *2012 12th International Conference on Hybrid Intelligent Systems (HIS)*, IEEE, pp.300–305.

Topaloglu, S. and Ozkarahan, I. (2011) 'A constraint programming-based solution approach for medical resident scheduling problems', *Computers & Operations Research*, Vol. 38, No. 1, pp.246–255.

Trick, M.A. and Yildiz, H. (2011) 'Benders' cuts guided large neighborhood search for the traveling umpire problem', *Naval Research Logistics (NRL)*, Vol. 58, No. 8, pp.771–781.

Tricoire, F., Romauch, M., Doerner, K.F. and Hartl, R.F. (2010) 'Heuristics for the multi-period orienteering problem with multiple time windows', *Computers & Operations Research*, Vol. 37, No. 2, pp.351–367.

Tsiligirides, T. (1984) 'Heuristic methods applied to orienteering', *Journal of the Operational Research Society*, Vol. 35, No. 9, pp.797–809.

Vansteenwegen, P., Souffriau, W., Berghe, G.V. and Oudheusden, D.V. (2009) 'Iterated local search for the team orienteering problem with time windows', *Computers & Operations Research*, Vol. 36, No. 12, pp.3281–3290.

Vansteenwegen, P., Souffriau, W. and Oudheusden, D.V. (2011) 'The orienteering problem: a survey', *European Journal of Operational Research*, Vol. 209, No. 1, pp.1–10.

Verbeeck, C., Vansteenwegen, P. and Aghezzaf, E-H. (2017) 'The time-dependent orienteering problem with time windows: a fast ant colony system', *Annals of Operations Research*, Vol. 254, Nos. 1–2, pp.481–505.

Vilím, P. (2007) *Global Constraints in Scheduling*, PhD thesis, Department of Theoretical Computer Science and Mathematical Logic, Charles University in Prague.

Vincent, F.Y., Jewpanya, P., Ting, C-J. and Redi, A.P. (2017) 'Two-level particle swarm optimization for the multi-modal team orienteering problem with time windows', *Applied Soft Computing*, Vol. 61, No. 1, pp.1022–1040.

Vincent, F.Y., Jewpanya, P., Lin, S-W. and Redi, A.P. (2019) 'Team orienteering problem with time windows and time-dependent scores', *Computers & Industrial Engineering*, Vol. 127, No. 1, pp.213–224.

Wari, E. and Zhu, W. (2019) 'A constraint programming model for food processing industry: a case for an ice cream processing facility', *International Journal of Production Research*, Vol. 57, No. 21, pp.6648–6664 [online] https://doi.org/10.1080/00207543.2019.1571250.