Computers & Industrial Engineering 107 (2017) 178–195

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

A constraint programming approach for the team orienteering problem with time windows

Ridvan Gedik [a,*], Emre Kirac [b], Ashlea Bennet Milburn [c], Chase Rainwater [c]

[a] Department of Mechanical and Industrial Engineering, University of New Haven, 300 Boston Post Rd, West Haven, CT 06516, USA
[b] Department of Engineering, University of Houston - Clear Lake, 2700 Bay Area Blvd., Houston, TX 77058, USA
[c] Department of Industrial Engineering, University of Arkansas, 4207 Bell Engineering Center, 1 University of Arkansas, Fayetteville, AR 72701, USA

ARTICLE INFO

Article history:
Received 26 December 2015
Received in revised form 14 February 2017
Accepted 13 March 2017
Available online 16 March 2017

Keywords:
Team orienteering problem
Time windows
Vehicle routing
Constraint programming
Interval variable

ABSTRACT

The team orienteering problem with time windows (TOPTW) is a NP-hard combinatorial optimization problem. It has many real-world applications, for example, routing technicians and disaster relief routing. In the TOPTW, a set of locations is given. For each, the profit, service time and time window are known. A fleet of homogenous vehicles are available for visiting locations and collecting their associated profits. Each vehicle is constrained by a maximum tour duration. The problem is to plan a set of vehicle routes that begin and end at a depot, visit each location no more than once by incorporating time window constraints. The objective is to maximize the profit collected. In this study we discuss how to use constraint programming (CP) to formulate and solve TOPTW by applying interval variables, global constraints and domain filtering algorithms. We propose a CP model and two branching strategies for the TOPTW. The approach finds 119 of the best-known solutions for 304 TOPTW benchmark instances from the literature. Moreover, the proposed method finds one new best-known solution for TOPTW benchmark instances and proves the optimality of the best-known solutions for two additional instances.

1. Introduction

The team orienteering problem with time windows (TOPTW) is a NP-hard combinatorial optimization problem (Labadie, Mansini, & Melechovsk, 2012). The formal definition of the TOPTW is as follows. Assume that $G = (\mathcal{N}, \mathcal{A})$ is a directed network graph with a set of $n+1$ nodes $\mathcal{N} = \{0, 1, \ldots, n\}$ and set of connecting arcs $\mathcal{A} = \{(i,j) : i \in \mathcal{N}, j \in \mathcal{N}, i \neq j\}$. The travel time t_{ij} on arc (i,j) is known. Associated with each location $i \in \mathcal{N}$, service time s_i, profit p_i, and time window $[b_i, e_i]$, where b_i and e_i are the earliest and latest times i can be visited, respectively, are known. If a vehicle visits location i and arrives there before b_i, it must wait until b_i to begin service. The profit p_i is collected if there is a visit to i within $[b_i, e_i]$. A fleet of m homogenous vehicles is available. The problem is to determine a set $\mathcal{V}$ of vehicle tours where each customer is visited *at most once* and each tour v starts and ends at the depot ($i = 0$) within window $[b_0, e_0]$. The objective is to maximize the profit collected from visited customers. The TOPTW is an extension of the more general orienteering problem (OP), first introduced in Tsiligirides (Tsiligirides, 1984). The OP considers only a single vehicle, while the TOPTW utilizes multiple vehicles and includes time window constraints (Chao, Golden, & Wasil, 1996). A comprehensive review of applications and solution techniques for OP variants, including TOPTW, is provided in Vansteenwegen, Souffriau, and Oudheusden (2011). Example applications include tourist routing problems (Souffriau & Vansteenwegen, 2010; Souffriau, Vansteenwegen, Vertommen, Berghe, & Oudheusden, 2008; Sylejmani, Dorn, & Musliu, 2012), disaster relief logistics (Kirac, 2016; Rath & Gutjahr, 2014), pickup and delivery services (Gutiérrez-Jarpa, Marianov, & Obreque, 2009), and sales representative route planning (Tricoire, Romauch, Doerner, & Hartl, 2010).

Many heuristic solution techniques have been developed for TOPTW in recent years. Montemanni and Gambardella (2009) propose an ant-colony system (ACS) algorithm for TOPTW. They also propose and solve 148 benchmark instances for TOPTW, which they develop by modifying vehicle routing problem with time windows (VRPTW) instances from Solomon (1987) and Cordeau, Gendreau, and Laporte (1997). In Gambardella, Montemanni, and Weyland (2012), the ACS algorithm is improved and better solutions are obtained for 26 TOPTW benchmark instances. Vansteenwegen, Souffriau, Berghe, and Oudheusden (2009) develop an easy to implement iterated local search (ILS) heuristic for TOPTW. While ILS is faster than the original ACS algorithm (Montemanni & Gambardella, 2009), the solutions from ACS are

* Corresponding author.
E-mail address: rgedik@newhaven.edu (R. Gedik).

http://dx.doi.org/10.1016/j.cie.2017.03.017
0360-8352/

approximately 2% better, on average, than solutions produced by ILS. Tricoire et al. (2010) develop a variable neighborhood search (VNS) heuristic for a variation of TOPTW; namely, the multi-period orienteering problem with multiple time windows. When the VNS heuristic is used to solve the TOPTW benchmark instances, it is observed that solution quality is better than ILS but computational time is worse. The solution quality and computational time of VNS is better than ACS. Lin and Yu (2012) develop fast and slow versions of a simulated annealing (SA) heuristic. The slow SA heuristic has longer computational times than the fast one, but is able to find best-known solutions for more instances than the fast SA heuristic. For both the slow and fast SA heuristics, the solution quality is better than ILS but worse than ACS and VNS. Labadie et al. (2012) develop a granular variable neighborhood search (GVNS) algorithm based on linear programming. The solution quality of GVNS is better than ILS and ACS but worse than VNS. GVNS is faster than ACS and VNS but slower than ILS. Qian and Andrew (2014) develop an iterative three-component heuristic (I3CH) that finds improved solutions for 35 of the 304 TOPTW benchmark instances. The first component of I3CH is local search, the second is a simulated annealing algorithm, and then finally routes are recombined to obtain better solutions. Cura (2014) develops an artificial bee colony (ABC) algorithm for TOPTW. The solution quality of ABC is worse than I3CH and GVNS but better than ACS. It is able to produce high-quality solutions with shorter runtime. On average, the computational time of ABC is better than I3CH, GVNS and ACS.

There is only one exact approach for TOPTW in the literature of which we are aware. Tae and Kim (2015) introduce a branch and price algorithm capable of solving both the team orienteering problem (TOP) and TOPTW. Of the three sets of TOPTW benchmarks available in the literature, they include only those from Righini and Salani (2009) in their computational study. They state the instances from Montemanni and Gambardella (2009) are too difficult to solve optimally, and also omit those from Vansteenwegen et al. (2009) because the optimal solutions of the instances are already known. For the 117 TOPTW benchmark instances included in their computational study, the branch and price algorithm finds optimal solutions for 91 of them within a two-hour runtime limit.

In this paper, we propose a new exact solution technique for the TOPTW. We formulate TOPTW using a constraint programming (CP) model and refer to this model as CP-TOPTW. We use CP Optimizer with two different branching rules for its solution. CP has been shown to be an efficient solution technique for numerous combinatorial optimization problems. Applications in the literature include problems such as parallel machine scheduling (Edis & Oguz, 2012; Gedik, Rainwater, Nachtmann, & Pohl, 2016; Hooker, 2007; Jain & Grossmann, 2001; Nachtmann, Mitchell, Rainwater, Gedik, & Pohl, 2014), tournament organization (Trick & Yildiz, 2011), staff scheduling & rostering (He & Qu, 2012; Topaloglu & Ozkarahan, 2011), vehicle routing & traveling salesman problems (Pesant, Gendreau, Potvin, & Rousseau, 1998, 1999; Quoc & Anh, 2010), and VRPTW (De Backer, Furnon, Shaw, Kilby, & Prosser, 2000; Guimarans, Herrero, Ramos, & Padrón, 2013; Rousseau, Gendreau, & Pesant, 2002; Rousseau, Gendreau, Pesant, & Focacci, 2004; Shaw, 1998). Using CP Optimizer with our model outperforms the branch and price approach of Tae and Kim (2015) in two primary ways. First, we solve the 187 and 66 TOPTW benchmark instances from Montemanni and Gambardella (2009) and Vansteenwegen et al. (2009), respectively, that Tae and Kim (2015) omit. We find solutions with a competitive average gap (2% and 0.24%) for those instances. Second, the branch and price approach fails to find a feasible solution within a two-hour runtime limit for 28 of the 117 TOPTW benchmark instances included in the Tae and Kim (2015) computational study. Using CP Optimizer, we find at least one feasible solution within a 30-min runtime limit for each of these 117 instances. On the other hand, one strength of the branch and price approach is that optimality is proven for more of the 117 instances than we are able to prove using CP Optimizer.

The contributions of this paper are threefold. First, a CP model is introduced for TOPTW and CP Optimizer with two branching rules is used for its solution. Due to the strengths of CP in expressing complex relationships, very difficult constraints such as selective node visits, subtour elimination and time windows are represented. Compared with ILP formulations for TOPTW, CP-TOPTW does not require a large number of decision variables and constraints. Thus, we are able to run benchmark instances without experiencing any memory problems. When compared with approximate approaches in the literature such as sophisticated local search methods, our CP model does not require extensive parameter tuning as those methods do. And while the approximate methods are quite efficient in finding good quality solutions, they are not able to prove the optimality of those solutions, as we are able for some instances using CP. Second, CP-TOPTW and its components, such as global constraints, provide a strong base for other solution techniques for OP variants and related routing problems, potentially fostering new methodological developments. Third, the results we obtain using CP Optimizer with CP-TOPTW advance current knowledge regarding TOPTW benchmark instances in a number of ways. In keeping with the convention in the literature, we use the term *best-known solution* to refer to a feasible solution with objective value equal to the maximum objective value published in the literature. We find 119 of the previously best-known solutions and we improve upon the best-known solution for one benchmark instance, finding a solution with an objective value strictly greater than what is published in the literature. For the 66 instances for which optimal solutions are known, we find 49 of them. Additionally, we provide new proof of optimality for two test instances.

The remainder of this paper is organized as follows. Section 2 provides the CP formulation for TOPTW and provides an illustrative example. Section 3 provides results for CP-TOPTW and a comparison to existing algorithms from the literature. Finally, conclusions and future research directions are discussed in Section 4.

2. Methodology

Vansteenwegen et al. (2009) discuss the computational difficulties associated with solving the TOPTW. It is known that solving TOPTW in polynomial time is unlikely (Lin & Yu, 2012). To address these computational challenges, we aim to test the effectiveness of CP, which is well known for its abilities to express complex relationships using global constraints and to obtain good quality solutions within reasonable times. A CP implementation contains a search strategy and a constraint propagation mechanism designed to filter out the values in (integer) variable domains that cause infeasible solutions (Hooker, 2006; van Hoeve & Katriel, 2006). In the constraint model, algorithms are triggered every time a change occurs in the domain of a variable. A feasible solution is obtained when all domains are reduced to a single value. Note that a variable can be used to model more than one constraint. Therefore, whenever a change occurs in the domain of a shared variable, propagation algorithms of all global constraints are run to search for the possible domain reductions of other variables (Harjunkoski & Grossmann, 2002; Lombardi & Milano, 2012; van Hoeve & Katriel, 2006). If a feasible solution has not been achieved after all possible reductions, value instantiation takes place. If all variables are instantiated and a constraint is not satisfied, then a