



The travelling salesman problem with time windows: Adapting algorithms from travel-time to makespan optimization[☆]



Manuel López-Ibáñez^{a,*}, Christian Blum^{b,c}, Jeffrey W. Ohlmann^d, Barrett W. Thomas^d

^a IRIDIA, CoDE, Université Libre de Bruxelles, Brussels, Belgium

^b IKERBASQUE, Basque Foundation for Science, Bilbao, Spain

^c Department of Computer Science and Artificial Intelligence, University of the Basque Country, San Sebastian, Spain

^d Department of Management Sciences, University of Iowa, Iowa City, USA

ARTICLE INFO

Article history:

Received 17 May 2012

Received in revised form 1 April 2013

Accepted 21 May 2013

Available online 7 June 2013

Keywords:

Ant colony optimization

Compressed simulated annealing

Travelling salesman problem with time windows

Hybridization

ABSTRACT

In combinatorial optimization it is not rare to find problems whose mathematical structure is nearly the same, differing only in some aspect related to the motivating application. For example, many problems in machine scheduling and vehicle routing have equivalent formulations and only differ with respect to the optimization objective, or particular constraints. Moreover, while some problems receive a lot of attention from the research community, their close relatives receive hardly any attention at all. Given two closely related problems, it is intuitive that it may be effective to adapt state-of-the-art algorithms—initially introduced for the well-studied problem variant—to the less-studied problem variant. In this paper we provide an example based on the travelling salesman problem with time windows that supports this intuition. In this context, the well-studied problem variant minimizes the travel time, while the less-studied problem variant minimizes the makespan. Indeed, the results show that the algorithms that we adapt from travel-time minimization to makespan minimization significantly outperform the existing state-of-the-art approaches for makespan minimization.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

As in the classical travelling salesman problem (TSP), the travelling salesman problem with time windows (TSPTW) requires the specification of a Hamiltonian cycle through a graph of nodes, but adds the requirement that each node must be visited within a predefined time window. For a routing application, the TSPTW represents the problem of finding an efficient route, starting and ending at a specified depot, that visits a set of customers, each one in his predefined time window. In the machine scheduling environment, the TSPTW can be used to model the problem of sequencing jobs on a single machine where the setup time of each job depends on the previous job, and each job has a release time and a deadline. In the routing context, the TSPTW objective is typically to minimize the sum of the travel times. In the machine scheduling context, the TSPTW objective is to minimize the makespan. Both versions of the TSPTW are proven to be NP-hard, and even finding

feasible solutions is an NP-complete problem [1]. For convenience we henceforth refer to the TSPTW with travel-time minimization by TSPTW-TT, and to the TSPTW with makespan minimization by TSPTW-M.

1.1. Contribution of this paper

Among the two above-mentioned problem versions, the TSPTW-TT has been more closely studied in the literature. In fact, very few algorithms have been proposed for the TSPTW-M. In this paper, we adapt two state-of-the-art algorithms for the TSPTW-TT, compressed annealing (CA) [2] and the Beam-ACO algorithm [3], to the TSPTW-M. These algorithms, by virtue of being metaheuristics, can be easily adapted to different variants of the same problem. Moreover, in this case, the structure of the problem remains the same despite the change in the objective function. We do not attempt to adapt other algorithms developed for the TSPTW-TT to the TSPTW-M. First, some of these algorithms heavily rely on heuristics specific to the TSPTW-TT. Second, since Beam-ACO and CA are the state-of-the-art for the TSPTW-TT, there is no a priori indication that other algorithms could become better than them when adapted to the TSPTW-M.

Concerning the experimental results, we present an exhaustive comparison between both adapted algorithms on a comprehensive set of benchmark instances available in the literature. This, as such,

[☆] This work was supported by grant TIN2012-37930-C02-02 of the Spanish Government. Manuel López-Ibáñez acknowledges support from the Belgian F.R.S.-FNRS, of which he is a postdoctoral researcher.

* Corresponding author. Tel.: +32 02650 2745.

E-mail addresses: manuel.lopez-ibanez@ulb.ac.be (M. López-Ibáñez), christian.c.blum@gmail.com (C. Blum), jeffrey-ohlmann@uiowa.edu (J.W. Ohlmann), barrett-thomas@uiowa.edu (B.W. Thomas).

is a useful contribution because existing algorithm proposals for the TSPTW-M have only been applied to rather small subsets of the available benchmark instances. Furthermore, both algorithms are compared to current state-of-the-art algorithms developed specifically for the TSPTW-M.

Concerning the comparison between Beam-ACO and CA, we were able to detect—for all considered sets of benchmark instances—a statistically significant advantage of Beam-ACO over CA. With respect to the comparison of Beam-ACO and CA with existing state-of-the-art algorithms, the following conclusions can be drawn. First, for the benchmark set proposed by Dumas et al. [7], Beam-ACO is able to improve the best-known results from [12] in five out of 22 cases. In the remaining cases, the same results as in [12] are achieved. Currently, the state-of-the-art algorithms for the TSPTW-M are ACS-TSPTW and ACS-Time [16]. Both algorithms are based on the metaheuristic ant colony optimization. In [16], both algorithms were applied to the benchmark set proposed by Potvin and Bengio [22]. A comparison of Beam-ACO and CA with the two ACO algorithms on this benchmark set reveals that Beam-ACO outperforms ACS-TSPTW in 17 out of 24 cases and ACS-Time in 20 out of 24 cases. While in the remainder cases both algorithms obtain the best-known solution in every run, Beam-ACO is from 40 to more than one hundred times faster than the other algorithms. The results of CA are similar to those of Beam-ACO.

1.2. Organization of the paper

The remainder of the paper is organized as follows. In Section 2, we provide a technical description of the TSPTW. We review the history of both problem variants in Section 2.1. In Section 3, we briefly describe CA and Beam-ACO as well as providing a description of the changes that were necessary for adapting them to makespan optimization. The benchmark instances considered in this work are described in Section 4. After outlining the tuning process, we provide comprehensive results in Section 5 for both algorithms on a comprehensive set of benchmark instances used in the related literature. Finally, we offer conclusions and an outlook to future work in Section 6.

2. Technical problem description

Given an undirected complete graph $G=(N, A)$, where $N=\{0, 1, \dots, n\}$ is a set of nodes representing the depot (node 0) and n customers, and $A=N \times N$ is the set of edges connecting the nodes, a TSPTW solution is a sequence visiting each node of G exactly once (starting and ending at the depot). We represent a tour as $P=(p_0=0, p_1, \dots, p_n, p_{n+1}=0)$, where the sub-sequence $(p_1, \dots, p_k, \dots, p_n)$ is a permutation of the nodes in $N \setminus \{0\}$ and p_k denotes the index of the node at the k th position of the tour. The two additional elements—that is, $p_0=0$ and $p_{n+1}=0$ —represent the depot at which the tour starts and ends.

For each edge $a_{ij} \in A$, which connects the two nodes i and j , there is an associated cost $c(a_{ij})$. In a routing application, $c(a_{ij})$ represents the service time at customer i plus the travel time between customer i and j . In a scheduling application, $c(a_{ij})$ represents the task time for job i plus the time to set up for job j when following job i . A time window $[e_i, l_i]$ is associated to each node $i \in N$, which defines when service at node i can begin. Waiting times are generally permitted; in the routing, this implies that a customer i may be reached before the start e_i of its time window, but service cannot start until e_i . Therefore, given a particular tour P , the departure time from node p_k is calculated as

$$D_{p_k} = \max(A_{p_k}, e_{p_k}), \tag{1}$$

where $A_{p_k} = D_{p_{k-1}} + c(a_{p_{k-1}, p_k})$ is the arrival time at node p_k in the tour. A tour P is feasible, if and only if $\Omega(P) = \sum_{k=0}^{n+1} \omega(p_k) = 0$, where $\omega(p_k) = 1$ if $A_{p_k} > l_{p_k}$, and 0 otherwise.

As outlined above, two different objective functions may be considered for the TSPTW. The function for travel-time minimization, $f_{tt}(\cdot)$, concerns the minimization of the sum of the costs of the edges traversed along a tour. That is, given a tour P :

$$f_{tt}(P) = \sum_{k=0}^n c(a_{p_k, p_{k+1}}) \tag{2}$$

Note that the objective function $f_{tt}(\cdot)$ is analogous to the one of the classical TSP. The other alternative is to minimize $A_{p_{n+1}}$, that is, the arrival time at the depot. That is, given a tour P :

$$f_m(P) = A_{p_{n+1}} \tag{3}$$

where function $f_m(\cdot)$ refers to makespan minimization.

2.1. Previous work on the TSPTW

The earliest papers on the TSPTW focused on exact approaches for makespan optimization [4,5]. The testing of these approaches was limited to instances of at most 50 nodes. Moreover, these algorithms were not able to handle wide or overlapping time windows. Langevin et al. [6] proposed a branch-and-bound scheme to solve a two-commodity flow formulation for the TSPTW and considered both makespan and travel-time optimization; their implementation was able to solve instances of up to 40 nodes. Dumas et al. [7] extended earlier dynamic programming approaches by using state space reduction techniques that allowed them to solve larger problem instances. More recently, Ascheuer et al. [8] developed a branch-and-cut algorithm in which they applied techniques tailored for the asymmetric TSPTW. Balas and Simonetti [9] presented a linear-time dynamic programming algorithm for several TSP variants with precedence constraints, including the TSPTW. Constraint programming is another exact approach that has been applied to the TSPTW [10,11].

Due to the inherent difficulty of the TSPTW, heuristics have been the focus of research in more recent years. Carlton and Barnes [12] developed a tabu search approach to optimize a hierarchical objective that attempts to primarily minimize makespan, and then secondly to minimize total travel time subject to maintaining the minimal makespan. In order to deal with infeasibility, Carlton and Barnes [12] augment the objective function with a static penalty for violating l_i of customer i . Gendreau et al. [13] presented a constructive heuristic with a subsequent improvement procedure. [14] presented a constructive heuristic that starts with a solution to an ad-hoc assignment problem, proceeds with a greedy insertion procedure to obtain a complete solution and applies local search to further improve the obtained solution. Two state-of-the-art TSPTW-TT algorithms for travel-time optimization were proposed by Ohlmann and Thomas [2] and López-Ibáñez and Blum [3]. Ohlmann and Thomas [2] proposed a compressed annealing (CA) algorithm, which is a variant of simulated annealing making use of a variable penalty method. Finally, López-Ibáñez and Blum [3] presented Beam-ACO algorithm, which results from the combination of ant colony optimization (ACO) with beam search, a heuristic variant of branch-and-bound. The same authors [3] also provided a comprehensive comparison of CA and Beam-ACO. Recently, a variable neighborhood search approach has been proposed [15]. This algorithm appears to perform similarly well. However, it has not been tested on all available benchmark instances. Authors developing heuristic approaches have focused on travel-time optimization and fewer results are published on makespan optimization. Curiously, TSPTW-M has been the focus of the ant colony optimization community [16–18]. Some of these ACO approaches [16,17] can be considered state-of-the-art for the TSPTW-M.

متن کامل مقاله

دریافت فوری ←

ISIArticles

مرجع مقالات تخصصی ایران

- ✓ امکان دانلود نسخه تمام متن مقالات انگلیسی
- ✓ امکان دانلود نسخه ترجمه شده مقالات
- ✓ پذیرش سفارش ترجمه تخصصی
- ✓ امکان جستجو در آرشیو جامعی از صدها موضوع و هزاران مقاله
- ✓ امکان دانلود رایگان ۲ صفحه اول هر مقاله
- ✓ امکان پرداخت اینترنتی با کلیه کارت های عضو شتاب
- ✓ دانلود فوری مقاله پس از پرداخت آنلاین
- ✓ پشتیبانی کامل خرید با بهره مندی از سیستم هوشمند رهگیری سفارشات