ARTICLE IN PRESS
JID: EOR [m5G;September 12, 2019;5:43]
European Journal of Operational Research xxx (xxxx) xxx
Contents lists available at ScienceDirect
European Journal of Operational Research
journal homepage: www.elsevier.com/locate/ejor

Interfaces with Other Disciplines

Shortest path tour problem with time windows

Luigi Di Puglia Pugliese[a], Daniele Ferone[b], Paola Festa[c], Francesca Guerriero[a,*]

[a] Department of Mechanical, Energetic and Management Engineering, University of Calabria, Italy
[b] Department of Informatics, Systems and Communication (DISCo), University of Milano-Bicocca, Italy
[c] Department of Mathematics and Applications, University of Napoli Federico II, Italy

ARTICLE INFO

Article history:
Received 24 July 2018
Accepted 28 August 2019
Available online xxx

Keywords:
Networks
Shortest path tour problem
Resource-constrained shortest path problem
Time windows constraints

ABSTRACT

This paper aims at studying a new variant of the shortest path tour problem, where time window constraints are taken into account. This is the first work dealing with the shortest path tour problem with time windows. The problem is formally described and its theoretical properties are analyzed. We prove that it belongs to the NP-hard class of complexity by polynomial reduction from the knapsack problem. An optimal solution approach based on the dynamic programming paradigm is devised. Labelling algorithms are defined along with well-tailored pruning strategies based on cost and time. The correctness of the bounding strategies is proven and the empirical behavior is analyzed in depth. In order to evaluate the performance of the proposed approach, extensive computational experiments have been carried out on a significant set of test problems derived from benchmarks for the shortest path tour problem. Sensitivity analysis is carried out by considering both algorithmic and instance parameters.

1. Introduction

The *Shortest Path Tour Problem* (SPTP) has been firstly introduced by Bajaj (1971) as a constrained version of the *Shortest Path Problem* (SPP). The problem is formulated on a graph and the nodes are clustered in several ordering sets. A cost is associated with each arc of the graph. The aim is to find a path with minimum cost, such that the sets are served in the order they are defined. In particular, a set is said to be served if at least one node belonging to it appears in the path. The optimal solution can be composed by subpaths of not served nodes, that play the role of connections between two consecutive served nodes.

One of the main applications of *SPTP* can be found in the domain of Network Functions Virtualization (NFV). NFV decouples the network service functionality from the underlying network, computer, and storage resources, and allows communication services to be composed by stitching together functional building blocks that may not be co-located and may be offered by different providers. The orchestration is the process of arranging and coordinating multiple network services to deliver the desired functionality. Orchestration relies on a “marketplace of services”: a repository of services and network functionalities that are available to users. An NFV marketplace planner has to construct a path from source to destination that visits virtual nodes where instances of these services have been deployed (Bhat, 2017).

Although the *SPTP* was introduced more than 40 years ago in Bajaj (1971), it has not been studied for a long time. Its resolution is proposed as an exercise of Bertsekas’s Dynamic Programming and Optimal Control book (Bertsekas, 2005), where it is asked to formulate it as a dynamic programming problem.

To the best of our knowledge, the first consistent work on *SPTP* is Festa (2012), in which it is proved that *SPTP* belongs to **P** complexity class, reducing it to the *SPP*. Several algorithms for its solution are proposed and some correlations with the Uncapacitated Facility Location Problems (*UFLP*) are explored. Later, Festa, Guerriero, Laganá, and Musmanno (2013) proposed a dynamic programming based labeling algorithm, and Bhat and Rouskas (2017) developed a depth-first tour search algorithm that outperformed the state of the art.

Several different variants of the problem have been proposed. Ferone, Festa, Guerriero, and Laganá (2016b) proposed a variant of *SPTP*, named *Constrained SPTP* (*CSPTP*), in which the solution path *P* can cross each arc of the input graph at most once. They proved that *CSPTP* is an **NP**-hard problem and proposed a Branch & Bound (B&B) algorithm to optimally solve it and a GRASP meta-heuristic. Independently, de Andrade and Saraiva (2018) and Ferone, Festa, and Guerriero (2019a) proposed two similar mathematical models. The former solved the problem using CPLEX, the latter proposed an efficient B&B.

* Corresponding author.
E-mail addresses: luigi.dipugliapugliese@unical.it (L. Di Puglia Pugliese), daniele.ferone@unimib.it (D. Ferone), paola.festa@unina.it (P. Festa), francesca.guerriero@unical.it (F. Guerriero).

https://doi.org/10.1016/j.ejor.2019.08.052
0377-2217/

Please cite this article as: L. Di Puglia Pugliese, D. Ferone and P. Festa et al., Shortest path tour problem with time windows, European Journal of Operational Research, https://doi.org/10.1016/j.ejor.2019.08.052

Finally, Carrabs, Cerulli, Festa, and Laureana (2017) proposed the *Forward SPTP*, in which it is possible to visit a node in T_k if and only if at least a node of each previous cluster, $T_1, \ldots, T_{k-1}$, has been already visited. They proposed a polynomial algorithm that is able to manage also the classical *SPTP*.

In this paper, we analyze a time-constrained version of the *SPTP* that includes time windows associated with each served node (*SPTPTW*). In this version of the problem, traversing time is associated with each arc and a feasible path must guarantee that the time needed to reach a served node does not exceed the due-date defined by the time window constraint.

Routing problems with time windows are widely studied in the scientific literature. The shortest path problem with time windows (*SPPTW*) was addressed for the first time in Desrosiers, Pelletier, and Soumis (1983) as a subproblem of a route construction problem. It was designed a label correcting method that generalizes the approach proposed by Gallo and Pallottino (1988) for the *SPP*.

Powell and Chen (1998) presented a label-correcting algorithm extending for the *SPPTW* the rule applied to the *SPP* by Glover, Glover, and Klingman (1984). Later, Desrochers and Soumis (1988) presented a label-setting algorithm that exhibits a pseudopolynomial complexity. The *SPPTW* arises as subproblem when the vehicle routing problem with time windows is solved via column-generation approaches (Liberatore, Righini, & Salani, 2011; Qureshi, Taniguchi, & Yamada, 2009; Tagmouti, Gendreau, & Potvin, 2007). For more details on the *SPPTW* and the related solution strategies, the reader is referred to Di Puglia Pugliese and Guerriero (2013a).

The *SPTPTW* shows some similarities with the *Generalized Vehicle Routing Problem with Time Windows* (GVRPTW) (Moccia, Cordeau, & Laporte, 2012), an extension of the *Generalized Vehicle Routing Problem* (GVRP) (Ghiani & Improta, 2000) since a clustered substructure for the nodes is considered. However, in the GVRPTW, the set of nodes is partitioned in sets of customers, and each set must be visited (served) exactly once. Moreover, since the sets are not ordered, the decisions involved in the problem are node selection (which node in each set is visited) and node sequencing (the sequence in which customer sets are served).

In the case of the *SPTPTW*, even though each subset must be served exactly once, it is possible to traverse each subset more than once (more details are given in Section 2). In addition, the start and end location of the vehicle can be different and, finally, the most important difference is that the service order of the subsets is given as input in the *SPTPTW*. For this reason, the decision about the node sequencing must not be taken.

To the best of our knowledge, this is the first time that tour constraints are included in the well-studied *SPPTW*. The *SPTPTW* has several applications. For instance, in tourism path planning, a tourist wants to visit a set of interesting areas, such as parks, museums, historical building, and squares. She/he can group these areas in several sets giving an order of visits to those sets. In each set, alternative places are collecting. The aim is to provide a tour, where one place belonging to each set is visited following the order imposed by the sets. In this context, one has to take into account the opening of the closing time of the interesting areas and the time needed to reach the target place in order to guarantee the maximum satisfaction from the planned tour.

Another example is cargo delivery. A vehicle must deliver orders to N regions in specific time windows, and the orders can be ranked respect to a priority. In this case, the regions can be represented as node subsets, and the priority is given by the order of the subsets.

The paper is organized as follows. In Section 2, the problem is formally described and theoretical properties are analyzed. Section 3 describes an optimal solution approach based on dynamic programming strategy along with tailored bounding techniques. Section 4 shows the computational results carried out considering several network topologies. The behavior of the proposed solution approach is analyzed in depth by reporting sensitivity analysis varying both algorithmic and network parameters. Section 5 concludes the paper providing some directions for future research.

2. Problem definition

Let $G(V, A)$ be a directed graph, where V is the set of n nodes and $A = \{(i, j) : i, j \in V\}$ is the set of m arcs. The set V contains two special nodes: the source node s and the destination node d. A non-negative cost c_{ij} and a non-negative transit time t_{ij} is associated with each arc $(i, j) \in A$. Let N denote a certain number of node subsets $T_1, \ldots, T_N$, such that $T_h \cap T_k = \emptyset$, for all $h, k = 1, \ldots, N$, $h \neq k$. Without loss of generality, we assume that $T_1 = \{s\}$ and $T_N = \{d\}$. Let $\mathcal{T} = \bigcup_{h=1}^{N} T_h$. Moreover, a non-negative service time s_i and a time window $[e_i, l_i]$ are associated with each node $i \in \mathcal{T}$, where e_i and l_i are the earliest and the latest arrival time, respectively. Given two distinct nodes i_1 and i_ν, a path $\pi_{i_1, i_\nu} =< i_1, \ldots, i_\nu >$ is an ordered sequence of nodes from i_1 to i_ν, such that $(i_w, i_{w+1}) \in A$, $w = 1, \ldots, \nu - 1$. The cost $c(\pi_{i_u, j_\nu})$ of a path π_{i_u, j_ν} is defined as the sum of the cost associated with its arcs, i.e. $c(\pi_{i_u, j_\nu}) = \sum_{w=u}^{\nu-1} c_{i_w, i_{w+1}}$.

The *SPTP* aims at finding the shortest path π_{sd} from the origin node $s \in V$ to the destination node $d \in V$ in the directed graph G, such that it serves successively and sequentially the subsets T_h, $h = 1, \ldots, N$ at the minimum cost. Sets T_h, $h = 1, \ldots, N$, must be served in exactly the same order in which they are defined. Let τ_{i_w} be the arrival time at node $i_w \in \pi_{sd}$, $w = 1, \ldots, |\pi_{sd}|$.

A path π_{sd} is said to be a feasible solution for the *SPTPTW* if the following conditions hold

$$\exists\, g_1, g_2, \ldots, g_N : g_1 < g_2 < \ldots < g_N, \\ i_{g_1} \in \pi_{sd} \cap T_1, i_{g_2} \in \pi_{sd} \cap T_2, \ldots, i_{g_N} \in \pi_{sd} \cap T_N; \tag{1}$$

$$e_{i_{g_h}} \le \tau_{i_{g_h}} \le l_{i_{g_h}}, \forall\, h = 1, \ldots, N. \tag{2}$$

Let Π be the set of paths from node $s \in V$ to the destination node $d \in V$ in the graph G.

Definition 2.1. The *SPTPTW* consists in finding a path $\pi^*_{sd} \in \Pi$ satisfying conditions (1) and (2) and corresponding to the minimum cost, i.e., such that $c(\pi^*_{sd}) \le c(\pi_{sd})$, $\forall\, \pi_{sd} \in \Pi$.

It is worth observing that the *SPTPTW* differs from the well-known *SPPTW* because a node i belonging to some set T_h could be either served or not. Therefore, there are not forbidden nodes in any moment. It is possible to traverse a node of a set T_i whenever it is needed, but the sets must be served in the exactly order of the sets $T_1, \ldots, T_N$. Indeed, condition (1) imposes that it must exist at least a sequence of strictly ascending indices $g_1, \ldots, g_N$, and each index denotes the node in the solution in which the set T_{g_i} is served. In other words, it is admitted both (i) to serve T_i and pass through it later in the solution and (ii) to traverse a node in T_i but serving it later in the solution. The constraint only imposes that it is possible to serve a given T_i after serving the set T_{i-1} and before serving T_{i+1}. Condition (2) ensures that the node chosen to serve a set T_{g_h} is visited within its associated time window. It is noteworthy that the time windows are not associated to the visit of nodes but to the sets service. Each node of the graph can be visited in any moment, but if it is used to serve the belonging set T_i, then the visit must occur within the time window. Let us consider the toy example depicted in Fig. 1. In this simple instance, all the transit times are equal to 1, and all the service times are equal to 1, but $s_2 = 3$. The costs are reported on the arcs. The graph of Fig. 1 contains two paths from $s = 1$ to $d = 7$, named