Computers and Operations Research 93 (2018) 66–78

Contents lists available at ScienceDirect

Computers and Operations Research

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

A branch-and-price algorithm for the Minimum Latency Problem

Teobaldo Bulhões [a], Ruslan Sadykov [b], Eduardo Uchoa [c,*]

[a] *Instituto de Computação, Universidade Federal Fluminense, Brazil*
[b] *Inria Bordeaux - Sud Ouest, France*
[c] *Departamento de Engenharia de Produção, Universidade Federal Fluminense, Brazil*

ARTICLE INFO

Article history:
Received 29 May 2017
Revised 19 January 2018
Accepted 20 January 2018

Keywords:
Minimum latency
ng-paths
Branch-and-price

ABSTRACT

This paper deals with the Minimum Latency Problem (MLP), a variant of the well-known Traveling Salesman Problem in which the objective is to minimize the sum of waiting times of customers. This problem arises in many applications where customer satisfaction is more important than the total time spent by the server. This paper presents a novel branch-and-price algorithm for MLP that strongly relies on new features for the *ng*-path relaxation, namely: (1) a new labeling algorithm with an enhanced dominance rule named multiple partial label dominance; (2) a generalized definition of *ng*-sets in terms of arcs, instead of nodes; and (3) a strategy for decreasing *ng*-set sizes when those sets are being dynamically chosen. Also, other elements of efficient exact algorithms for vehicle routing problems are incorporated into our method, such as reduced cost fixing, dual stabilization, route enumeration and strong branching. Computational experiments over TSPLIB instances are reported, showing that several instances not solved by the current state-of-the-art method can now be solved.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

This paper deals with the Minimum Latency Problem (MLP). In MLP, we are given a complete directed graph $G = (V, A)$ and a time t_{ij} for each arc $(i, j) \in A$. Set V is composed of $n + 1$ nodes: node 0, representing a depot, and nodes $1, \ldots, n$, representing n customers. The task is to find a Hamiltonian circuit $(i_0 = 0, i_1, \ldots, i_n, i_{n+1} = 0)$, a.k.a. tour, in G that minimizes $\sum_{p=1}^{n+1} l(i_p)$, where the latency $l(i_p)$ is defined as the accumulated travel time from the depot to i_p. The MLP is related to the Time Dependent Traveling Salesman Problem (TDTSP), a generalization of the Traveling Salesman Problem (TSP) in which the cost for traversing an arc depends on its position in the tour. More precisely, MLP can be viewed as the particular case of the TDTSP where the cost of an arc (i, j) in position p, $0 \le p \le n$, is given by $(n - p + 1)t_{ij}$.

The MLP is also known in the literature as Delivery Man Problem (Roberti and Mingozzi, 2014), Traveling Repairman Problem (Afrati, Foto et al., 1986), Traveling Deliveryman Problem (Tsitsiklis, 1992) and Traveling Salesman Problem with Cumulative Costs (Bianco et al., 1993). Although MLP seems to be a simple variant of TSP, some important characteristics are very different in those problems. First, two different viewpoints of a distribution system are considered: TSP is server oriented, since one wants to minimize the total travel time; on the other hand, MLP is customer oriented because the objective is equivalent to minimizing the average waiting time of customers (Archer and Williamson, 2003; Silva et al., 2012; Sitters, 2002). Customer satisfaction is the main objective in many applications, such as home delivery services (Méndez-Díaz et al., 2008), and has attracted the attention of researchers, as reflected by the considerable number of MLP variants studied in the very last years (see, for instance, Lysgaard and Wøhlk (2014),Rivera et al. (2016),Nucamendi-Guillén et al. (2016)and Sze et al. (2017)). Second, in contrast to what happens in TSP, simple local changes may affect globally a MLP solution because the latency of subsequent customers may change (Silva et al., 2012; Sitters, 2002). This can make it more difficult to solve MLP both exactly and heuristically. For example, current state-of-the-art exact methods for MLP are not capable of solving consistently instances with 150 customers, whereas TSP instances with thousands of customers are solved routinely (Abeledo et al., 2013).

Many complexity results for MLP have been obtained. The problem is NP-Hard for general metric spaces (Sahni and Gonzalez, 1976), and remains NP-Hard even if the times correspond to Euclidean distances (Afrati, Foto et al., 1986) or if they are obtained from an underlying graph that is a tree Sitters (2002). On the other hand, the problem is polynomial if the underlying graph is a path (Afrati, Foto et al., 1986; Garca et al., 2002), a tree with equal weights or a tree with diameter at most 3 (Blum et al., 1994).

* Corresponding author.
E-mail addresses: tbulhoes@ic.uff.br (T. Bulhões), Ruslan.Sadykov@inria.fr (R. Sadykov), uchoa@producao.uff.br (E. Uchoa).

https://doi.org/10.1016/j.cor.2018.01.016
0305-0548/© 2018 Elsevier Ltd. All rights reserved.

The MLP with deadlines, i.e., with upper bounds on $l(i_p)$, is NP-Hard even for paths (Afrati, Foto et al., 1986). In terms of approximation, hardness results show that one should not expect to attain arbitrarily good approximation factors for MLP (Blum et al., 1994). However, 3.59 and 3.03 approximations are known for general metric spaces and general trees, respectively (Archer and Blasiak, 2010; Chaudhuri et al., 2003). Moreover, a constant factor approximation is not likely to exist if times do not satisfy the triangle inequality, just as for TSP (Blum et al., 1994).

The first integer programming formulations were given in Picard and Queyranne (1978), where the authors stated TDTSP as a machine scheduling problem and solved instances with up to 20 jobs by means of a branch-and-bound method over lagrangian bounds. A new formulation with n constraints was presented in Fox et al. (1980), but the authors did not report any computational results. Lucena (1990) and Bianco et al. (1993) followed the same approach as Picard and Queyranne (1978) and employed langragian bounds in experiments over MLP instances with up to 30 and 60 vertices, respectively. The latter authors also developed a dynamic programming method capable of attesting that the bounds obtained for 60-vertex instances were within 3% from optimality. Then, a series of enumerative strategies based on new formulations was introduced in Fischetti et al. (1993), Van Eijl (1995), Méndez-Díaz et al. (2008), Bigras et al. (2008), Godinho et al. (2014), as well as cutting planes (Bigras et al., 2008; Méndez-Díaz et al., 2008; Van Eijl, 1995) and polyhedral studies (Méndez-Díaz et al., 2008). Instances with 60 vertices could already be solved by the algorithm of Fischetti et al. (1993). More recently, Abeledo et al. (2013) managed to solve almost all TSPLIB instances with up to 107 vertices using a branch-cut-and-price algorithm. The authors departed from a formulation by Picard and Queyranne (1978) and proposed new inequalities, that are proved to be facet-inducing. Roberti and Mingozzi (2014) implemented dual ascent and column generation techniques to compute a sequence of lower bounds associated with set partitioning formulations where a column represents an ng-path, which is a path relaxation introduced by Baldacci et al. (2011). An ng-path may contain cycles, but just those allowed by the so-called ng-sets. These sets are iteratively augmented so that less cycles are allowed and improved bounds are obtained. The final lower bound is used in a dynamic programming recursion to compute the optimal solution. This method could solve some larger TSPLIB instances, with up to 150 vertices, and currently holds the status of state-of-the-art exact method for MLP. Finally, heuristic algorithms for MLP can be found in Ngueveu et al. (2010),Salehipour et al. (2011),Silva et al. (2012)and Mladenović et al. (2013).

This paper presents a novel branch-and-price algorithm for MLP that strongly relies on ng-paths. Following the directions of Roberti and Mingozzi (2014), our method works over a set partitioning formulation where columns represent ng-paths and the column generation bounds computed on each node of the tree are derived from dynamically defined ng-sets. However, we introduce the following improvements on the use of ng-paths.

- **Multiple Partial Label Dominance:** In the labeling algorithms used for pricing ng-paths, a partial path P is represented as a label $L(P)$. A key concept in this kind of algorithm is dominance. A label $L(P_1)$ dominates a label $L(P_2)$ if every completion P' of P_2 is also a feasible completion of P_1, and the cost of $P_1 + P'$ is not larger than the cost of $P_2 + P'$. In this case, $L(P_2)$ can be safely eliminated. In this paper, we propose a stronger dominance rule by which some extensions for $L(P_2)$ can be avoided, even though this label cannot be completely disregarded according to the classical dominance rule. We briefly discuss two alternative implementations of this new dominance rule, where the best one typically speeds up the labeling algorithm by factors between 4 and 8.
- **Arc-Based ng-Path Relaxation:** ng-sets as originally defined by Baldacci et al. (2011) are a vertex-based memory mechanism. In this paper, we provide a generalized definition of them in terms of arcs. We show that this new definition is particularly useful in the context of dynamically defined ng-sets, allowing strong bounds to be obtained in more controlled pricing times.
- **Fully Dynamic ng-Path Relaxation:** We improve the dynamic ng-path relaxation of Roberti and Mingozzi (2014) by introducing a procedure for decreasing the ng-sets, without changing the current bounds. Such reductions are beneficial for the pricing time and also help to refine the choice of ng-sets.

Also, other well-known elements of efficient exact algorithms for many other variants of the vehicle routing problem (VRP) are incorporated into our method, namely reduced cost fixing, dual stabilization, route enumeration and strong branching. Computational experiments over MLP instances derived from TSPLIB were conducted to attest the effectiveness of the new branch-and-price algorithm. The results show that better bounds can be obtained in less computational time when compared to the state-of-the-art algorithm, especially because of the new features for the ng-path relaxation. In particular, the branch-and-price solved all the 9 instances with up to 150 vertices not solved in Roberti and Mingozzi (2014). It could also solve 4 additional instances, with more than 150 vertices, never considered before by exact methods.

The remainder of this paper is organized as follows. Section 2 discusses the ng-path relaxation and labeling algorithms. Section 3 introduces the new features for the ng-path relaxation. The proposed branch-and-price algorithm is described in Section 4, where we also give implementation details. Computational experiments are presented in Section 5. Finally, concluding remarks are drawn in the last section.

2. Route relaxations and labeling algorithms

This section reviews the route relaxations and labeling algorithms that are related to current state-of-the-art exact algorithms for VRPs, such as Capacitated VRP (CVRP), VRP with time windows (VRPTW), and the MLP itself. Such algorithms are based on a combination of column and cut generation over the following set-partitioning formulation.

$$\min \sum_{R\in\Omega} c_R \lambda_R \tag{1}$$

$$\text{s.t.} \quad \sum_{R\in\Omega} a_R^i \lambda_R = 1, \qquad \forall i \in \mathcal{C}, \tag{2}$$

$$\lambda_R \in \{0,1\}, \qquad \forall R \in \Omega, \tag{3}$$

where $\mathcal{C}$, Ω, c_R and a_R^i denote, respectively, the set of customers, the set of feasible routes, the cost of route R, and the number of times route R visits customer i.

As the number of variables in Formulation (1)–(3) is exponential in $|\mathcal{C}|$, column generation is typically applied to solve its linear relaxation. The pricing subproblem depends on the considered variant, but it can often be modeled as the Elementary Resource Constrained Shortest Path Problem (ERCSPP). In ERCSPP, we are given a directed graph $G' = (V', A')$ with vertex set V' and arc set A'; source and sink nodes $s \in V'$ and $t \in V'$, respectively; and a set of resources $\mathcal{W}$. Moreover, for each $i \in V'$ and $r \in \mathcal{W}$, let $l_i^r \in \mathbb{R}$ and $u_i^r \in \mathbb{R}$ be, respectively, the minimum and the maximum consumption of resource r in any partial path from s to i. Each partial path $P = (i_0 = s, i_1, \ldots, i_p)$ has an associated vector of resource

Download English Version:

https://daneshyari.com/en/article/6892665

Download Persian Version:

https://daneshyari.com/article/6892665

Daneshyari.com