Operations Research Letters 41 (2013) 211–215

Contents lists available at SciVerse ScienceDirect

Operations Research Letters

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

Efficient feasibility testing for request insertion in the pickup and delivery problem with transfers

Renaud Masson, Fabien Lehuédé, Olivier Péton *

LUNAM Université, Ecole des Mines de Nantes, IRCCyN UMR CNRS 6597 (Institut de Recherche en Communications et Cybernétique de Nantes), Nantes, France

ARTICLE INFO

Article history:
Received 24 September 2012
Received in revised form
15 January 2013
Accepted 16 January 2013
Available online 23 January 2013

Keywords:
Pickup and delivery problem
Transfers
Feasibility check
Forward time slack

ABSTRACT

The Pickup and Delivery Problem with Transfers (PDPT) consists of defining a set of minimum cost routes in order to satisfy a set of transportation requests, allowing them to change vehicles at specific locations. In this problem, routes are strongly interdependent due to request transfers. Then it is critical to efficiently check if inserting a request into a partial solution is feasible or not. In this article, we present a method to perform this check in constant time.

1. Introduction

The Pickup and Delivery Problem with Transfers (PDPT) [3] consists of defining a set of routes that minimize the distance traveled in order to satisfy a set of transportation requests. Each transportation request has a known origin called the *pickup point* and a destination called the *delivery point*. The PDPT allows for transfers (also referred to as preemption in other papers): a request may be picked up by one vehicle, dropped at a transfer point and later be moved to its delivery point by another vehicle. Several requests can share a vehicle as long as its capacity is not exceeded. The service at pickup and delivery points must occur during time windows.

Neighborhood-based heuristics are common methods for solving vehicle routing problems with time-related constraints. It is well known that a critical factor of performance is the ability of the method to quickly check the feasibility of a solution. For example, Hunsaker and Savelsbergh discussed a sufficient condition to assess the feasibility of a route in the dial-a-ride problem under maximum wait time and maximum ride time constraints [6]. They show that this sufficient condition can be checked in linear time. Firat and Woeginger show that the whole feasibility check can be performed in linear time [5]. Feasibility testing raises an additional difficulty in the case of the PDPT where routes are *interdependent*. Transferring a request from one vehicle to another introduces precedence constraints at the transfer point. Since in many heuristic methods, millions of insertions are evaluated during the search, their feasibility must be checked as efficiently as possible.

The previous heuristic algorithms developed for the PDPT did not discuss the complexity of their feasibility tests [7,8].

Contributions of this note. In the PDPT, checking the feasibility of a request insertion contains two parts. The first part checks the temporal constraints of the problem. We propose an incremental method which enables us to check in constant time if a given insertion is consistent with the temporal constraints of the problem. This method is based on the *Forward Time Slack* principle of Savelsbergh [9]. The second part checks the vehicle capacity constraints. Since this test is similar to those performed in the pickup and delivery problem with no transfer, we do not discuss it. The present paper covers only the PDPT, but we believe this work is relevant for a variety of vehicle routing problems with precedence or synchronization constraints (see [4] for a survey).

In many pickup and delivery applications, the time of departure or arrival of vehicles may be imposed. Without loss of generality, we assume that all vehicles leave the depot at time 0 and all requests are serviced as early as possible. Our approach can be easily adapted to the symmetric case, with imposed arrival date at the depot and requests serviced as late as possible.

The remainder of the paper is organized as follows: The PDPT is formally stated in Section 2. In Section 3 we recall the Forward Time Slack principle. Section 4 defines the feasibility problem. Section 5 extends the Forward Time Slack principle to the PDPT. Section 6 presents the feasibility test for an insertion without transfer, while Section 7 concerns the insertion of a

* Correspondence to: Ecole des Mines de Nantes, 4 rue Alfred Kastler, F-44307 Nantes Cedex 3, France. Tel.: +33 251 858 313; fax: +33 251 838 349.
E-mail addresses: renaud.masson@mines-nantes.fr (R. Masson), fabien.lehuede@mines-nantes.fr (F. Lehuédé), olivier.peton@mines-nantes.fr (O. Péton).

0167-6377/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
doi:10.1016/j.orl.2013.01.007

request through a transfer point. The feasibility check described in this article requires some preprocessing which is described in Section 8: each time a request is removed from or inserted into a current solution, the service times of the requests and a matrix of waiting times must be updated. This update has a quadratic complexity but it is much less frequent than the constant time feasibility check described here.

2. The pickup and delivery problem with transfers

The PDPT can be formally described as follows. Let R be the set of transportation requests, T the set of transfer points and K the set of homogeneous vehicles of capacity Q. A request $r \in R$ corresponds to the transportation of q_r load units from a pickup point p_r to a delivery point d_r. The sets of all pickup and delivery points are denoted by P and D respectively. Each vehicle performs a single route. The starting and ending depots of vehicles are designated by o and o' respectively. The PDPT is defined on a complete directed graph $G = (V, U)$ where $V = P \cup D \cup T \cup \{o\} \cup \{o'\}$ and $U = \{(u, v) | u, v \in V, u \neq v\}$. With each arc $(u, v) \in U$ is associated a non-negative travel time $\theta_{u,v}$. We assume that all travel times respect the triangle inequality. A time window $[e_v, l_v]$ is associated with each vertex $v \in V$, where e_v and l_v represent the earliest and the latest time at which the service can begin at vertex v. The service at each vertex $v \in P \cup D \cup T$ has a known duration s_v, modeling the time needed to get requests in or off the vehicle. The number of load units carried simultaneously by a vehicle cannot exceed its capacity Q.

A vehicle can wait at a vertex only if it arrives there before the opening of the corresponding time window. If two requests have a common pickup or delivery point, the corresponding vertex is duplicated. According to this modeling, a vertex cannot be associated with more than one request.

A solution of the PDPT is a set of $|K|$ routes serving all requests such that each vehicle starts at o and ends at o'. For every request $r \in R$, vertices p_r and d_r can be served by the same route, provided that p_r is served before d_r. Vertices p_r and d_r can also be served by distinct routes $k \in K$ and $k' \in K$. In this case k and k' have to visit a common transfer point $\tau \in T$, such that τ is visited after p_r in route k and before d_r in route k'. Moreover, r must get off vehicle k at τ before being reloaded into k'. The objective of the PDPT is to serve all requests at minimal cost.

3. Forward time slack

We recall the principle of Forward Time Slack proposed by Savelsbergh [9] for the Traveling Salesman Problem with Time Windows (TSPTW) and the Vehicle Routing Problem with Time Windows (VRPTW).

Let us first introduce some notation that is used all along the paper: let u and v be two vertices serviced by a route k and $\psi_{u,v}$ the ordered set of vertices on the path $(u, \ldots, v)$. The waiting time at a vertex $i \in \psi_{u,v}$ is denoted by w_i. The direct predecessor and successor of vertex i in route k are denoted by $\rho(i)$ and $\sigma(i)$ respectively. The set of all successors of i is denoted by $\Gamma(i)$.

We assume that the service times are scheduled as early as possible. Thus the service time h_v at v is equal to the service time h_u, plus the travel and service durations along the path, plus the waiting times at the vertices of the path. If we denote $TWT_{\psi_{u,v}} = \sum_{i \in \psi_{\sigma(u),v}} w_i$, the total waiting time on path $(u, \ldots, v)$, then

$$h_v = h_u + \sum_{i \in \psi_{u,\rho(v)}} (s_i + \theta_{i,\sigma(i)}) + TWT_{\psi_{u,v}}. \quad (1)$$

For each vertex u on a route k, Savelsbergh introduces the notion of *forward time slack* F_u to represent the maximal amount of time the service at vertex u can be postponed without violating any time window at its successors:

$$F_u = \min_{i \in \{u\} \cup \Gamma(u)} \left\{ l_i - h_u - \sum_{j \in \psi_{u,\rho(i)}} (s_j + \theta_{j,\sigma(j)}) \right\}.$$

Using forward time slacks, the feasibility of a client insertion in a route can be checked in constant time. As pointed out by Cordeau et al. [1], based on Eq. (1), this equation can be rewritten as

$$F_u = \min_{i \in \{u\} \cup \Gamma(u)} \left\{ TWT_{\psi_{\sigma(u),i}} + l_i - h_i \right\}. \quad (2)$$

According to [9], the feasibility of 2-exchanges and or-exchanges for the TSPTW and the VRPTW can be checked in constant time if the variable $TWT_{\psi_{u,v}}$ is known for any pair of vertices (u, v) in the considered route.

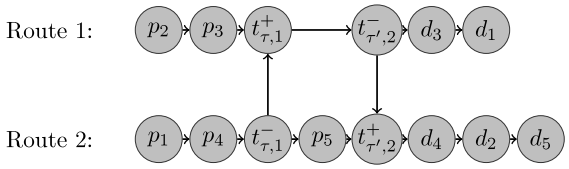

Fig. 1. Example of a precedence graph for two routes interconnected at transfer points τ and τ'.

4. Insertion feasibility for the PDPT

We consider a partial solution s of the PDPT that satisfies the temporal constraints of the problem. Let us denote by $R_s \subset R$ the subset of requests served in solution s. The sets of pickup and delivery vertices in this solution are denoted by P_s and D_s respectively. For each request $r \in R_s$ transferred at a transfer point $\tau \in T$, this point is split into an inbound transfer vertex $t^-_{\tau,r}$, where the request gets off the vehicle, and an outbound transfer vertex $t^+_{\tau,r}$, where the request gets in another vehicle. The sets of inbound and outbound transfer vertices used in s are denoted by T^-_s and T^+_s respectively. A partial solution of the PDPT can be associated with a *precedence graph*, defined below.

Definition 1 (*Precedence Graph*)**.** The precedence graph associated with a solution s of the PDPT is defined by $G_s = (V_s, A_s)$, where

$$V_s = P_s \cup D_s \cup T^-_s \cup T^+_s \cup \{o\} \cup \{o'\}$$

and A_s contains arcs (u, v) such that $v = \sigma(u)$ or u and v are the inbound and outbound transfer points of the same request.

An example of precedence graph is represented in Fig. 1 that models a partial solution with two routes and five requests. In this example, requests 1 and 2 are transferred at points τ and τ' respectively. This is modeled by four vertices: $t^-_{\tau,1}$, $t^+_{\tau,1}$, $t^-_{\tau',2}$ and $t^+_{\tau',2}$.

A *schedule* on G_s defines a service time h_i for each vertex $i \in V_s$. The schedule is *feasible* if the temporal constraints of the PDPT are satisfied for s. As stated before, in this paper we consider the vertices to be scheduled as early as possible and we denote by w_i the waiting time at vertex $i \in V_s$.

Let $r \in R \setminus R_s$ be a request to insert in this partial solution. Inserting r in s at a given position consists of inserting the corresponding vertices and arcs in the precedence graph G_s. This is feasible if the new solution resulting from this insertion has a feasible schedule.

A request r can be inserted without transfer or through a specified transfer point τ. As inserting without transfer is similar to and simpler than inserting with transfers, we provide the feasibility condition for the latter case only. Therefore we consider