European Journal of Operational Research 266 (2018) 29–34

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

Complexity of routing problems with release dates and deadlines

Damián Reyes[1,*], Alan L. Erera, Martin W. P. Savelsbergh

H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, United States

ARTICLE INFO

Article history:
Received 22 September 2016
Accepted 14 September 2017
Available online 21 September 2017

Keywords:
Routing
Dynamic delivery
Release dates and deadlines
Computational complexity
Dynamic programming

ABSTRACT

The desire of companies to offer same-day delivery leads to interesting new routing optimization problems. We study the complexity of single depot dispatching problems in which a delivery to a customer must occur within a pre-specified time after the customer places the order. Thus, each order has a release date (when the order can be dispatched from the depot) and a service guarantee that implies a deadline (when the order needs to be delivered). A vehicle delivering an order cannot depart the depot before the order is released, and must arrive at the customer at or before the order's deadline. We show that single and multiple vehicle variants where customers are located on a half-line can be solved to optimality in polynomial time. This setting, as well as our results, generalize those found in Archetti, Feillet, and Speranza (2015).

1. Introduction

In dynamic *delivery* problems, vehicles deliver goods locally from an origin depot (or, perhaps a small number of origin depots) to customer locations, and the requests for delivery arise during the vehicle operating period; e.g., Azi, Gendreau, and Potvin (2012). Although dynamic vehicle routing problems have received a significant amount of attention in the literature (see, for example, the surveys by Berbeglia, Cordeau, and Laporte (2010); Pillac, Gendreau, Guéret, and Medaglia (2013), and Psaraftis, Wen, and Kontovas (2016)), the class of dynamic delivery problems is just beginning to be explored. The defining characteristic in these problems is that when customer requests become known, they not only have a deadline, which specifies the latest time the delivery can be made, but also a *release date*, which specifies the earliest time the goods to be delivered are ready to be dispatched from the depot. The ready time can be the time that a request is made, but it can also be later to account, for example, for order picking and staging.

What makes vehicle routing problems with release dates interesting and challenging is the trade-off between delaying the dispatch of a vehicle until more requests are ready, in order to build a route that serves many customers and has a low delivery cost per order, and dispatching a vehicle early, in order to have more time prior to deadlines for the vehicle to travel and deliver orders, i.e., replacing nonproductive time at the depot waiting for orders to become ready with productive time on the road delivering orders. In dynamic settings, the challenge is even greater because of the uncertainty surrounding if and when future orders will be placed; delaying the dispatch of a vehicle increases the risk of being unable to meet the deadline of future requests.

Archetti, Feillet, and Speranza (2015) study the complexity of deterministic variants of some important classes of dynamic delivery problems. Specifically, they propose the Traveling Salesman Problem with release dates (TSP-rd) in which a single vehicle operates one or more consecutive (non-overlapping in time) routes from a single depot, each time loading and delivering released orders to customer locations, and the Uncapacitated Vehicle Routing Problem with release dates (UVRP-rd) which differs only in that the routes may overlap in time (because multiple vehicles are available to execute the routes). For each problem, the authors consider two variants: in one, the objective is to minimize total distance traveled while completing all delivery routes by a common deadline T, and in the other the objective is to minimize the latest completion time of any route. Not surprisingly, all of these optimization problems are *NP*-hard when customer locations are nodes in a general network. However, the authors show that each of these problems is solvable in polynomial time for problem instances where customer locations and the depot are all points in the "real line" metric space $\mathbb{R}$, with distance and time between $x, y \in \mathbb{R}$ given by $d(x, y) = |y - x|$.

Specifically, Archetti et al. (2015) develop dynamic programming algorithms for both variants of the TSP-rd problem that run in $O(n^3)$ time, and an $O(n^2)$ algorithm for UVRP-rd when minimizing distance traveled. In this paper, we propose alternative dynamic programming algorithms with a complexity of $O(n^2)$ for the

* Corresponding author.
E-mail addresses: ldrr3@gatech.edu (D. Reyes), alan.erera@isye.gatech.edu (A.L. Erera), mwps@isye.gatech.edu (M.W.P. Savelsbergh).
[1] skype: damian.reyes.

https://doi.org/10.1016/j.ejor.2017.09.020
0377-2217/

same two variants of the TSP-rd problem on the half-line. Furthermore, we extend the TSP-rd setting by considering *service guarantees*, which constrain each delivery to a customer to occur within a fixed amount of time S after the release date. We thus consider not only customer-specific release dates r_i, but also a customer-specific delivery deadlines $r_i + S$. Service guarantees are common in dynamic delivery contexts as retailers seek to offer instant gratification to consumers. For example, a retailer may guarantee online shoppers that an order will be delivered within two hours after the order is placed. Another environment where such a service guarantee arises naturally is meal delivery: "if your pizza is not delivered within 45 minutes after you place your order, it will be free." In addition to the alternative algorithms for the variants considered in Archetti et al. (2015), we propose dynamic programs for TSP-rd problems with service guarantees that (i) minimize the completion time of the last route and (ii) minimize the distance traveled while completing the last route by deadline T, running in $O(n^2)$ and $O(n^3)$ time for instances on the half-line, respectively. We also develop an $O(n^3)$ time algorithm for the UVRP-rd problem with service guarantees on the half-line that minimizes distance traveled.

The remainder of the paper is organized as follows. In Section 2, we formally introduce the problems studied and the notation used throughout. In Section 3, we prove a number of structural properties of optimal delivery schedules. In Section 4, we describe and analyze dynamic programming algorithms for single vehicle problems. In Section 5, we similarly describe and analyze a dynamic programming algorithm for the multiple vehicle problem minimizing travel distance. Section 6 contains some qualitative insights and concluding remarks.

2. A vehicle routing problem with release dates and order deadlines

In this section, we introduce an extension of the TSP-rd problem that includes order delivery deadlines. Let $N = \{1, \ldots, n\}$ be a set of customers located on the real half line $\mathbb{R}^+ \equiv [0, +\infty)$, and assume that the single depot is located at $x = 0$. Let the location of customer $i \in N$ be given by τ_i, and measure travel distances and times such that a round trip from the depot to i requires $2\tau_i$ distance and time. Note furthermore that a route traveling from the depot to customer i and back under these assumptions can also make deliveries to any set J of customers located such that $\tau_j \le \tau_i$ for $j \in J$ while still incurring $2\tau_i$ distance and time.

Suppose each customer $i \in N$ places an order with a release time of r_i, which implies the earliest possible time a vehicle dispatched to deliver at i can depart the depot. In the remainder, we will use "customer" and "order" interchangeably for simplicity. Let S be a common service guarantee applied to all orders, and therefore $r_i + S$ specifies the deadline time by which order i must be delivered at i.

Assume that deliveries to customers occur instantaneously upon vehicle arrival, and that all deliveries are made by the vehicle on the outbound journey from the depot. Thus, the latest possible dispatch time l_i from the depot such that the service guarantee at i is met is given by $l_i = r_i + S - \tau_i$. We restrict attention only to instances where $\tau_i \le S$; instances not meeting this condition are trivially infeasible. Furthermore, we also restrict attention to instances where the latest time by which the final delivery route must be completed, T, is such that $T \ge r_i + S + \tau_i$ for all $i \in N$. This condition essentially states that the company will only accept a customer order such that if it is completed by its deadline, the vehicle has time to return to the depot on time. In problems without individual service guarantees, as considered in Archetti et al. (2015), the equivalent is to ensure that $T \ge r_i + 2\tau_i$ to avoid trivially infeasible instances.

Throughout this paper, without loss of generality, we assume $r_i < r_{i+1}$ for $i < n$: since locations are on the half-line and delivery is instantaneous, if there were multiple orders released simultaneously, we need only consider the order furthest away from the depot (the rest of orders can always be delivered on time within the outbound journey of a feasible route to the furthest location).

The primary feasibility problem we study in this paper is a single vehicle problem defined as follows:

Problem 1. Is there a sequence of delivery routes that can be executed by a single driver, each starting and ending at the depot, such that each order $i \in N$ is dispatched at or after r_i and delivered at or before $r_i + S$, and the last route is completed at or before T?

If customers $i_1, i_2, \ldots, i_k$ are served in delivery route K, then the earliest dispatch time of the route, $r(K)$, is $\max\{r_{i_1}, r_{i_2}, \ldots, r_{i_k}\}$, the latest dispatch time of the route, $l(K)$, is $\min\{l_{i_1}, l_{i_2}, \ldots, l_{i_k}\}$, and the furthest order visited on the route, $\tau(K)$, is $\max\{\tau_{i_1}, \tau_{i_2}, \ldots, \tau_{i_k}\}$ (for convenience, the empty route has $r(\emptyset) = 0$, $\tau(\emptyset) = 0$, $l(\emptyset) = T$). Let the completion time of K, denoted by $c(K)$, be the time that the driver is back at the depot after serving orders in K. Observe that $r(K) \le l(K)$ is necessary for feasibility, and that if the driver is at the depot at time $r(K)$, there is no reason to delay the dispatch of the route.

Before proceeding with the analysis of the problem, we introduce the notion of *non-interlacing* routes and delivery schedules that contain only non-interlacing routes. Let a route K be an ordered set of customers visited on a single dispatch from the depot.

Definition 1. Two routes K_1 and K_2 with $\min\{i \mid i \in K_1\} < \min\{j \mid j \in K_2\}$ are non-interlacing if and only if $\max\{r_i \mid i \in K_1\} < \min\{r_j \mid j \in K_2\}$.

Note that two routes serving a single customer each are always non-interlacing. In other words, any pair of interlacing routes involves at least three customers.

Definition 2. A delivery schedule $\mathcal{S}$ of non-interlacing routes is a partition of N that can be characterized by the set of last customers in each route, i.e., $\mathcal{S} = \{i_1, i_2, \ldots, i_k, n\}$ with $1 \le i_1 \le i_2 \le \cdots \le i_k \le n$, indicating that orders $\{1, \ldots, i_1\}$ are delivered on the first route, orders $\{i_1 + 1, \ldots, i_2\}$ are delivered on the second route, etc.

3. Structural properties of feasible and optimal delivery schedules

We now discuss key properties of feasible and optimal delivery schedules. First, observe that if K_1 and K_2 are non-interlacing routes with $r(K_1) < r(K_2)$, i.e., K_1 can be dispatched before K_2, then if K_1 and K_2 appear consecutively in an optimal delivery schedule, K_1 will be dispatched before K_2. Next, consider the following proposition which shows that we can limit our attention to delivery schedules for Problem 1 that contain only non-interlacing routes.

Proposition 1. *Any feasible delivery schedule for a single driver can be transformed into a feasible delivery schedule with non-interlacing routes, and no increase in total travel time.*

Proof. Suppose a feasible schedule contains interlacing routes $K_1 \supseteq \{i, k\}$ and $K_2 \supseteq \{j\}$, where $r_i < r_j < r_k$. Without loss of generality, suppose that (i, j, k) constitutes the earliest such order triplet, or, more precisely, suppose that (i, j, k) is the lexicographic minimum among the set of order triplets defining interlacing routes. This implies that i is the first order released in K_1, j is the first order released in K_2, and k is the first order in K_1 released after any order in K_2. Now consider the following cases:

1. Suppose K_1 is dispatched before K_2, which implies that $c(K_1) < c(K_2)$.