Submitted to *Transportation Science*
manuscript (Please, provide the manuscript number!)

A probabilistic approach to pickup and delivery problems with time window uncertainty

Péter Györgyi
Institute for Computer Science and Control, Budapest, Hungary, gyorgyi.peter@sztaki.mta.hu

Tamás Kis
Institute for Computer Science and Control, Budapest, Hungary, kis.tamas@sztaki.mta.hu

In this paper we study a dynamic and stochastic pickup and delivery problem proposed recently by Srour, Agatz and Oppen. We demonstrate that the cost structure of the problem permits an effective solution method without generating multiple scenarios. Instead, our method is based on a careful analysis of the transfer probability from one customer to the other. Our computational results confirm the effectiveness of our approach on the data set of Srour et al.

Key words: dynamic and stochastic pickup and delivery problems, network flows

1. Introduction

In this paper we consider dynamic pickup and delivery problems with time window uncertainties as defined recently by Srour, Agatz, and Oppen (2016). In that model there is a transportation service provider that gets call from customers with exact pickup and drop-off locations, but with inaccurate estimations of the time windows for the transportations. The time windows of the service requests become known with certainty only after a second call from the customers, shortly before the service may start.

Srour, Agatz, and Oppen (2016) describe a couple of real-world scenarios where the above uncertainty is predominant. For instance, harbor pilots, who drive ships to berth, know the location of the ship, and also where it will berth, but the arrival time of the ship is often uncertain. A related problem is transportation of containers by tracks from pickup points to drop-off locations, where the exact time of releasing a container at the pickup terminal is not known exactly. They also mention transportation of patients after medical treatments from the hospital to home, where the exact completion time of the treatments is not known with certainty. A related application is on-demand chaffeur services, that drive home clients in their own cars after a party. We can extend this list by transportation tasks in a workshop, where semi-finished goods must be transported by fork-lifts, or autonomously guided vehicles between the machining cells, and the pickup and drop-off locations are perfectly known, but the time window of service is uncertain even if a schedule of the manufacturing operations is broadcasted in advance.

As Srour, Agatz, and Oppen (2016) noted, in their examples the customers can request the transportation service by giving the exact location of the pickup and drop-off locations, while providing the time window of starting the service only approximately, e.g., around 2pm. Then, when the customer has more information about its service requirements, it calls the service provider again telling the time window in which it expects the transportation to be started from the pickup to the drop-off location. Since the pickup and drop-off locations may be known well in advance, and also some estimation of the time window of starting the service is preannounced by the customers, the service provider may exploit this information to increase service level, and to reduce its costs.

The main result of this paper is a new algorithm that may help transportation service providers that operate in the above context to find better vehicle tours. Our method is based on estimating the expected operational costs, where missing a customer request is heavily penalized, and the other component is the total deadhead cost (operating empty while going to the next pickup location or to the depot). The novelty of our approach is that unlike Srour, Agatz, and Oppen (2016), we do not generate scenarios, and we solve only a single minimum cost flow problem at each decision point. Yet, our method seems to outperform their method, in which at each decision point, multiple scenarios (realization of time windows) are generated, and a MIP model of a vehicle routing problem over all the remaining jobs is solved for each scenario. We believe that the success of our approach is due to the cost-structure of the problem at hand, where the penalty of rejecting a customer request is very high compared to deadhead costs.

In Section 2 we review the related literature, and in Section 3 we give a formal description of the problem studied. Our method is presented in Section 4, and our computational results are summarized in Section 5, where we compare our method to that of Srour et al. We conclude the paper in Section 6.

2. Literature review

Dynamic pickup-and-delivery is a rapidly developing field of transportation science, which is certified by a series of recent review papers, see e.g., Berbeglia, Cordeau, and Laporte (2010), Pillac et al. (2013), Psaraftis, Wen, and Kontovas (2016). In Psaraftis (1988), a vehicle routing problem is characterized as *dynamic*, if the input of the problem is received and updated concurrently with the determination of the routes. Using the terminology of Berbeglia, Cordeau, and Laporte (2010), in this paper we focus on a *one-to-one* problem, where each request has an origin and a destination. In a *dynamic and stochastic problem*, some exploitable stochastic information is available about the dynamically revealed information (Pillac et al. 2013).

The problem studied in this paper has recently been proposed by Srour, Agatz, and Oppen (2016). In their model, each customer first preannounces its request, then confirms it at some later time, not much before the service actually should take place. In the preannouncement, the exact pickup and drop-off locations are provided along with an estimation of the pickup time by means of a time window. However, the preannounced time window can change in the future when the customer confirms its request. On the other hand, the distribution of the difference between the start (or end) of the preannounced and the confirmed time windows is known. The authors propose 4 methods to solve the dynamic problem. All the methods are based on solving a mixed-integer linear program (MIP) which models a (static) pickup and delivery problem with some of the customer requests. The MIP is similar to that of Yang, Jaillet, and Mahmassani (2004) devised for a track-load pickup and delivery problem. In the "Ignore" method, preannouncements are ignored and at any time only the confirmed requests are used to determine the tours of the vehicles. In the "Naïve" method, preannounced time windows are used until the customers confirm their requests, from which time on they are replaced by the confirmed ones. However, in the more advanced MTS-veh and MTS-seq methods, first multiple scenarios are generated for the realization of preannounced, but unconfirmed time windows, which are used along with the confirmed ones in the MIP models to be solved. Each scenario gives a routing of the vehicles, from which a sophisticated method synthesizes the final tours used until the next event occurs, when the entire planning procedure is repeated with updated information. The methods MTS-veh and MTS-seq differ in the procedure that synthesizes the final tours of the vehicles. The computational results of the authors show that the best method is MTS-seq, and we will compare our results to the best results of Srour, Agatz, and Oppen (2016). We emphasize that at each decision point as many MIPs have to be solved as the number of scenarios generated, which was set to 60 by Srour et al. The scenario-based approach finds its roots in the paper of Bent and Van Hentenryck (2004), who propose a method for a dynamic routing problem with time windows. In their method, multiple scenarios are generated containing the known requests, and also some possible future requests. Future requests are obtained by sampling their probability distribution. As Bent and Van Hentenryck (2004) states, their approach is a generalization and abstraction of that of Gendreau et al. (1999), who propose a parallel tabu search method and adaptive memory management to accommodate new customer requests, but without sampling. For the scenario-based approach, see also Hvattum, Løkketangen, and Laporte (2006).

The main novelty of the model of Srour, Agatz, and Oppen (2016) is that until the customers confirm their requests, only stochastic information is available on the desired service time windows, but the pickup and drop-off locations are known from the preannouncements. In contrast, in most of the previous work on dynamic vehicle routing problems, the dynamic data consists of the complete user requests, i.e., pickup and drop-off locations, along with the desired time windows are revealed together. Mitrović-Minić, Krishnamurti, and Laporte (2004) consider a dynamic pickup and delivery problem with time windows where no probabilistic information about future requests are known. Instead, they divide the time horizon into short and long term, and apply different objective functions for the two periods when inserting new customer requests into the tours of the vehicles.

Günlük et al. (2006) propose a complex method for continually reoptimizing the schedule of a fleet of vehicles and drivers to adapt it to the new or updated reservations. They maintain a foreground schedule, which is always feasible, and it is modified either by incorporating into it the output of the integer programming based optimization engine run periodically, or by a fast heuristic to respond to changes since the last run of the optimization engine.

Ferrucci, Bock, and Gendreau (2013) devise a pro-active real-time control approach for a dynamic vehicle routing problem in which dummy customer requests are generated based on historic data to anticipate future requests. The authors classify the quality of stochastic knowledge attainable from past request information, and they identify structural diversity as a crucial criterion.

Muñoz-Carpintero et al. (2015) propose a method based on evolutionary algorithms to solve a dial-a-ride problem, in which future requests are not known in advance, but the average service patterns from the past are taken into account to devise robust tours for the vehicles.

Probabilistic information is explicitly used in (Bertsimas and Van Ryzin 1991) and (Bertsimas and Van Ryzin 1993), in which a single and respectively a multiple vehicle routing problem is studied. Service requests arrive according to a Poisson process and are uniformly distributed in a service region. Optimal routing strategies are identified that minimize the average waiting time of the customers.

Ichoua, Gendreau, and Potvin (2006) study a dynamic vehicle routing problem, where the area served is divided into geographical zones, and also the planning time horizon is divided into periods. The requests are not known in advance, but the probability of receiving at least one customer request in a given geographical zone and time period can be calculated. This information is used in order to decide if a vehicle should stay in the same zone and wait for customer requests or move to another zone in the next period. The authors adapt the method of Gendreau et al. (1999) to determine the routing of the vehicles.

Ho and Haugland (2011) formulate and solve a dial-a-ride problem, where each customer request has a probability known by the service provider. For finding the routes of the vehicles, a local search, and a tabu search procedure are proposed, in which the next solution is chosen by selecting the best (non-tabu) neighbor of the current solution. The value of a solution is its expected cost, and a procedure is devised for finding the best neighbor in $O(n^5)$ time. Therefore, the computation time of a single iteration is $O(n^5)$, which is considerable if the number of customers n is large.

Albareda-Sambola, Fernández, and Laporte (2014) consider a multi-period vehicle routing problem with probabilistic information. In their model, the time horizon is divided into time periods, and for the current as well as for the future periods, the probability that the given period is in the time window of the customer is known. For the current period it is 0 or 1, but for future periods, it can be any value between 0 and 1. In each time period it is decided which customers to serve, and also the tours of the vehicles serving them are planned.

Waiting strategies constitute a very important part of a solver for dynamic pickup and delivery problems. Briefly, a waiting strategy determines where and how long a vehicle should wait on its tour before serving the next request. Mitrović-Minić and Laporte (2004) describe 4 distinct waiting strategies in the context of a pickup and delivery problem of courier companies offering same-day delivery. Besides drive-first, and wait-first, dynamic waiting, and advanced dynamic waiting are suggested. In both of the latter strategies, the routes of the vehicles are divided (dynamically) into service zones (defined by customers not too far away in time), and the strategies differ how they insert waiting periods while traveling in the service zones, and between service zones. The waiting strategies are evaluated in simulation experiments.

Branke et al. (2005) propose waiting strategies to increase the probability that a new customer request can be inserted in the tour of one the of vehicles. The authors propose a number of simple waiting strategies which are evaluated in a simulation framework along with a more advanced evolutionary algorithm.

Thomas (2007) studies a dynamic and stochastic vehicle routing problem in which a single vehicle serves a set of customers, who request service over time. No time windows are attached to the requests, but the tour of the vehicle must end by a given time limit. Each time when the vehicle finishes serving a client, the decision consists of whether to include the latest requests into the tour, or reject them, and whether to wait or go to the next accepted customer. The author analyses the single-customer case and proposes 5 distinct waiting heuristics. Anticipatory route selection is the topic of Thomas and White III (2004), where optimal policies for a single vehicle are suggested.

Vonolfen and Affenzeller (2016) consider a pickup and delivery problem with time windows. They use historical data to fine tune their intensity-based waiting strategy, in which the *intensity* of a pair of time and location is defined as the average transition time for requests in the historical

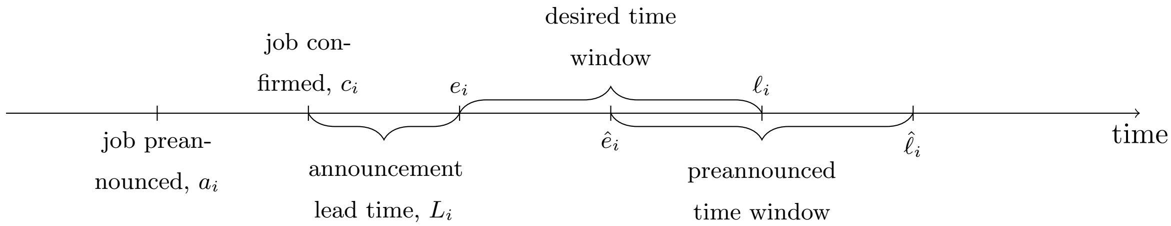


Figure 1 The various data attached to a request

request set that would have been not revealed yet. Previous and the proposed waiting strategies are compared in a computational study.

3. Problem statement

In our problem description, we closely follow that of Srour, Agatz, and Oppen (2016). The transportation service provider (service provider, for short) has a *fleet of vehicles*, V, and each vehicle can serve only one request at a time. The vehicles are identical from the point of view of the customers. The service provider receives a *sequence of pickup and delivery requests* over time from a set of customers J.

The customers first preannounce their service requests. The *preannouncement* for $i \in J$ is made at time a_i, and it specifies the pickup and the drop-off locations, along with an *estimation* of the earliest and latest pickup times, $\hat{e}_i$ and $\hat{\ell}_i$, respectively. Let $TW_i = \hat{\ell}_i - \hat{e}_i$ denote the length of the pickup time window. Then, each customer $i \in J$ *confirms* its request by calling the service provider at some time $c_i > a_i$ again, and specifying the *desired pickup time window* with the earliest pickup time e_i, and the latest pickup time $\ell_i = e_i + TW_i$. The difference $e_i - c_i$ is the *announcement lead time* L_i. The transportation service for customer i cannot start before e_i, or after ℓ_i. So, if no vehicle starts to serve customer i in the time window $[e_i, \ell_i]$, then the request is *rejected*. The above data is illustrated on a timeline in Figure 1.

The preannounced time window $[\hat{e}_i, \hat{\ell}_i]$ is only an estimation, or forecast of the desired time window $[e_i, \ell_i]$. The difference of $e_i - \hat{e}_i$ can be seen as a random variable known only in distribution in the course of planning until customer i confirms its request. The distribution may be empirically learned by the service provider operating for a longer period. So, we assume that e_i is uniformly distributed in $[\hat{e}_i - \Delta, \hat{e}_i + \Delta]$, for some known parameter Δ, see Srour, Agatz, and Oppen (2016). However, the announcement lead times are known in advance with certainty.

The profit earned by the service provider by serving a customer $i \in J$ is $\mathit{profit}_i = (f + \mathit{dist}_i \times g)$, where f and g are fixed amounts in some monetary unit, while dist_i is the distance between the pickup and the drop-off location of i. The *total cost* of the service provider is the total distance of the vehicles operating empty (moving from the depot to the first pickup location, from a drop-off location to the next pickup location, or back to the depot) multiplied by a cost factor h, the *routing* *cost*, denoted subsequently by RC, plus the *lost profit*, which is defined as $LP = \sum_{i \in J^{rej}} (f + dist_i \times g)$, where the summation is over all the rejected (unserved) customers J^{rej}, that is

$$total\ cost = RC + LP. \tag{1}$$

The cost of serving the requests, i.e., a function of the $dist_i$, is not added to the cost function, because that is payed by the customers, and we consider only the lost profit, and the cost of driving empty.

The vehicles start from a depot and have to return to the depot after finishing operation. At any moment of time, a vehicle can be in one of the following *states*: (i) waiting idle at some location (at the depot, at the pickup, or drop-off location of a customer, or at some waiting area), (ii) on the way to some target location set by the service provider, (iii) transporting a customer to its drop-off location. The service provider can interrupt (ii), and set a new target location for a vehicle, or may simply ask a vehicle to stop and wait at its current position until the next command. Like Srour, Agatz, and Oppen (2016), we assume that travel times of the vehicles are deterministic and can be calculated accurately using the distances between locations.

We want to suggest a strategy for the service provider that helps minimize the total cost (1). At any time moment the strategy knows all the preannounced, and confirmed requests, the announcement lead times along with the distribution of the possible realizations of the pickup time windows, and the states and current positions of the vehicles.

The above problem has recently been proposed by Srour, Agatz, and Oppen (2016), and in the next section we will describe an algorithm, which works well on their benchmark instances.

4. Algorithmic approach

In this section we describe a method that helps the transportation service provider to operate its vehicles. Firstly, we give an overview about the entire process in Section 4.1, and then we provide the core algorithm that has to be used repeatedly (Section 4.2).

4.1. Overview

The transportation service provider receives a sequence of pickup and delivery requests over time, and it maintains a routing plan for each vehicle under its control. The routing plans are adjusted time and again to take into account the new events. The vehicles get commands only for the next action.

New commands can be issued at any moment of time, and the current target location or state of a vehicle can be modified arbitrarily, with the exception that the transportation of a customer cannot be interrupted. In order to decide about the possible modification of the routing plans, the service provider has to solve an optimization problem while taking into account

- the state and the current position of the vehicles,
- the preannounced requests (pickup and drop-off locations, $[\hat{e}_i, \hat{\ell}_i]$), and the corresponding distribution of the possible realizations of the pickup time windows (e_i is uniformly distributed in $[\hat{e}_i - \Delta, \hat{e}_i + \Delta]$),
- the desired pickup time windows $[e_i, \ell_i]$ for the confirmed requests,
- the announcement lead times (L_i).

After solving the optimization problem, a subset of vehicles may receive new commands, i.e., if the result is that a vehicle has to change (i) its target location, or (ii) its state, then it gets a new command. Note that (i) may occur if a vehicle is on the way to a target location, but as a result of re-optimization, it has to go to another location, and (ii) may occur if the vehicle is waiting at some location, and the new routing plan sets a new target location, or it is on the way to some target location, and according to the new routing plan it has to stop at its current position and wait for the next command. As we will see in the computation results, waiting at some position may readily help to reduce the total distance traveled idle.

In the proposed strategy, re-optimization occurs upon any of the following events:

- a preannouncement is received from a customer,
- a customer confirms its request,
- a vehicle arrives to the target location set by the service provider (waiting area, pickup / drop-off location).

Events are processed in chronological order, no special tie-breaking rule is applied. In the next section we describe the optimization algorithm to determine the new routing plans for the vehicles.

4.2. Probabilistic estimation and min-cost-flows

In this section we describe the optimization problem solved by the service provider each time it wishes to adjust the routing plans of the vehicles.

Suppose that (re)optimization occurs at time t_{act}. We say that a customer $i \in J$ is *rejected* at time t_{act}, if it has already confirmed its request ($c_i \leq t_{act}$), it is not served yet, and the latest pickup time $\ell_i < t_{act}$.

The essence of the method is to build a network with a source node s and a sink node t, one node for each vehicle v, and two nodes, i^+ and i^-, for each customer (which has not started, finished, or rejected yet) representing the pickup and drop-off locations, respectively. There are directed arcs from the source node to the vehicle nodes, from the vehicle nodes to the pickup nodes of the customers, from the pickup to the drop-off node of the same customer, from the drop-off nodes of the customers to the pickup nodes of other customers, and from each vehicle node and from each drop-off node to the sink node (see Figure 2). Each $s - t$ path in this network represents a routing

v_k: vehicle node
i^+, i^-: pickup and drop-off nodes of customer i

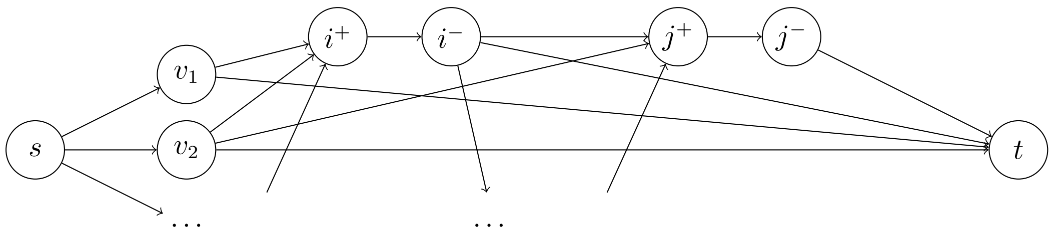


Figure 2 Fragment of the network

plan of a vehicle, i.e., the first node of the path after the source node is a vehicle node, then comes a (possibly empty) alternating sequence of pickup and drop-off nodes, and finally, an arc to the sink node representing the way back to the depot.

The cost of an arc leading to the pickup location of some customer represents the *expected* travel cost from a vehicle or from the drop-off location of another customer. By expected travel cost and expected profit we mean that with each arc there is associated a probability:

(a) the probability of an arc from some vehicle node v to a customer node i is the chance that v can arrive to i in the desired time window $[e_i, \ell_i]$ of i. If the request of customer i is not confirmed yet, then knowing the distribution of $e_i - \hat{e}_i$, and the actual state and position of the vehicle, this probability can be computed. Otherwise, if customer i has confirmed its request, this probability is 0 or 1, since a vehicle can either arrive to a customer (possibly after finishing the current transportation service for another customer), or not.

(b) the probability of an arc from some customer i to another customer j is the chance that a vehicle serving i can arrive to the pickup location of j after finishing i.

Note that the expected profit is subtracted from the expected travel cost, because then if we minimize the cost of the paths (by computing a minimum cost flow in the network), we minimize the travel costs and maximize the total profit earned, which is equivalent to (1), since $\sum_{i \in J}(f + g \times dist_i) = LP + \sum_{i \in J^{served}}(f + g \times dist_i)$, where J^{served} is the set of customers served.

Now we define the arc costs and capacities more formally. The supply of the source node s is set to $|V|$, which has to be carried to the sink node t, which has a matching demand. Each arc to be defined below has capacity 1. The source node is connected to each vehicle node by a directed arc of cost 0, and for each customer i, there is a directed arc of cost 0 from i^+ to i^-. Each vehicle node v is connected to each customer node i^+ provided the probability that vehicle v can serve customer i, denoted by $P(v \text{ } can \text{ } serve \text{ } i)$, is above a given threshold value. The cost of the arc (v, i^+) is $P(v \text{ } can \text{ } serve \text{ } i) \times (travel \text{ } cost \text{ } to \text{ } customer \text{ } i - profit_i)$. The travel cost to customer i depends on the state of the vehicle v. If v is transporting a customer to its drop-off location, then it is $h\times$(the distance from the drop-off location of the customer being served to the pickup location of customer i) (recall the definition of cost factor h in Section 3). Otherwise, it is $h\times$(the distance from the current position of the vehicle to the pickup location of i). The probability $P(v\ can\ serve\ i)$ will be defined later in this section. Each vehicle is also connected to the sink node t with a directed arc (v,t). The cost of this arc depends on the state of the vehicle. If v is transporting a customer to its drop-off location, then it is $h\times$(the distance from the drop-off location of the customer to the depot), otherwise it is $h\times$(the distance from the current position of v to the depot). For each pair of customers i,j, there is a directed arc from i^- to j^+ in the network provided the probability that the same vehicle can serve j after finishing i, $P(j$ can be served after $i)$, is above a given threshold value. The cost of the arc is $P(j$ can be served after $i)\times$((*travel cost from the drop-off location of customer* i *to the pickup location of customer* j) - *profit*$_i$). Finally, there is a directed arc from each customer node i^- to the depot with cost $h\times$(the distance from the drop-off location of i to the depot). For an illustration, see Figure 2.

PROPOSITION 1. *The minimum cost flow problem always admits an optimal integral (0/1) solution. Furthermore, the arcs with flow value 1 induce $|V|$ (internally) node disjoint $s-t$ paths and possibly isolated directed cycles comprising only customer nodes.*

Proof Since arc capacities are uniformly 1, and the network admits $|V|$ arc disjoint $s-t$ paths through the vehicle nodes, there always exists an optimal, integral (0/1), minimum cost $s-t$ flow in the network, see e.g., Ahuja, Magnanti, and Orlin (1993). Furthermore, any feasible, integral $s-t$ flow can be decomposed into a set of $|V|$ internally node disjoint $s-t$ paths, and possibly to some isolated cycles consisting of only customer nodes i^+ and i^-, because from each node i^+ there is a single outgoing arc (to node i^-) of unit capacity. This decomposition immediately provides the tours of the vehicles. Notice that an integral optimal solution cannot contain $s-t$ walks with loops, i.e., a sequence of consecutive edges from s to t with unit flow on each arc of the sequence that passes through an arc at least twice, because such a walk should contain an arc (i^+,i^-) at least twice for some customer i, which is impossible, because then the inflow at node i^+ would be at least two, while the outflow can only be 1 due to the unit capacity of the arc (i^+,i^-). Q.E.D.

Using the proposition, it is easy to determine the next action of each vehicle, we only have to find the outgoing arc from each vehicle node v with unit flow.

It remains to determine the probabilities $P(v\ can\ serve\ i)$ and $P(j$ can be served after $i)$. In order to determine these quantities, we have to define a number of parameters. Firstly, the *travel time* t_{vi} of a vehicle v to customer i is the total time needed for vehicle v to arrive to the pickup location of customer i. If v is transporting some customer $j\neq i$, then first it has to arrive to the drop-off location of j, then it has to move on to the pickup location of i. Hence, this time can clearly be determined. Otherwise, t_{vi} is just the travel time from the current position of v to the pickup location of customer i. Second, the travel time t_{ij} from some customer i to some customer j is the time needed to drive from the drop-off location of customer i to the pickup location of customer j.

Now we determine $P(v\ can\ serve\ i)$. We distinguish two cases:

1. If customer i has not confirmed its request yet, then we define the *earliest finish of the pickup time window* as $ef_i = \max\{t_{act} + L_i + TW_i, \hat{\ell}_i - \Delta\}$, and the *latest finish of the pickup time window* as $lf_i = \hat{\ell}_i + \Delta$, If $ef_i > lf_i$, then $P(v\ can\ serve\ i) = 0$. Likewise, if $t_{act} + t_{vi} > lf_i$, then vehicle v cannot serve customer i, and $P(v\ can\ serve\ i) = 0$. Otherwise,

$$P(v\ can\ serve\ i) = \frac{lf_i - \max\{t_{act} + t_{vi}, ef_i\}}{lf_i - ef_i}.$$

   Notice that if $lf_i = ef_i$, then both the numerator and the denominator are 0, and $P(v\ can\ serve\ i) = 0$.

2. If customer i has confirmed its request, then v can serve i only if $t_{act} + t_{vi} \leq \ell_i$. Hence, $P(v\ can\ serve\ i) = 1$ if $t_{act} + t_{vi} \leq \ell_i$, and 0 otherwise.

Next, we determine $P(j$ can be served after $i)$. To this end, we determine the earliest and latest finish time of serving customer i, and the earliest and latest pickup time of customer j. If i has already confirmed its request, then the earliest finish time of serving i is $ef_i = \max\{e_i, t_{act}) + t_i$, and the latest possible time to finish i is $lf_i = \ell_i + t_i$, where t_i is the travel time from the pickup location to the drop-off location of customer i. Otherwise, if i has only made the preannouncement, then $ef_i = \max\{t_{act} + L_i, \hat{e}_i - \Delta\} + t_i$, and $lf_i = \hat{\ell}_i + \Delta + t_i$.

If customer j has confirmed its request, then the earliest and also the latest time point when the pickup time window of customer j can end is $ep_j = lp_j = \ell_j$. Otherwise, $ep_j = \max\{t_{act} + L_j + TW_j, \hat{\ell}_j - \Delta\}$, and $lp_j = \hat{\ell}_j + \Delta$.

Let X_i be a random variable representing the completion time of serving customer i, and Y_j a random variable representing the value of ℓ_j. X_i and Y_j are considered independent. X_i takes values from the interval $[ef_i, lf_i]$, and Y_j from $[ep_j, lp_j]$. Moreover, unless $ep_j = lp_j$, Y_j is uniformly distributed on the interval $[ep_j, lp_j]$ with probability density function

$$f_{Y_j}(y) = \frac{1}{lp_j - ep_j}.$$

The distribution of X_i depends on TW_i. That is, let $w := \min\{TW_i, (lf_i - ef_i)/2\}$. Since the desired time window $[e_i, \ell_i]$ is uniformly distributed in $[\hat{e}_i - \Delta, \hat{\ell}_i + \Delta]$ by assumption, the probability

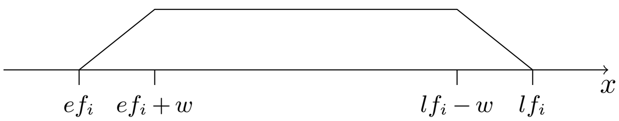


Figure 3 **An example of** $f_{X_i}(x)$

of *finishing* the request of customer i is less likely in the intervals $[ef_i, ef_i + w]$ and $[lf_i - w, lf_i]$, so we define the probability density function of X_i as follows:

$$f_{X_i}(x) = \begin{cases} \frac{x - ef_i}{w(lf_i - ef_i - w)} & \text{if } ef_i \le x \le ef_i + w, \\ \\ \frac{1}{lf_i - ef_i - w} & \text{if } ef_i + w \le x \le lf_i - w, \\ \\ \frac{lf_i - x}{w(lf_i - ef_i - w)} & \text{if } lf_i - w \le x \le lf_i. \end{cases} \tag{2}$$

It is easy to verify that $\int_{ef_i}^{lf_i} f_{X_i}(x)dx = 1$. A possible $f_{X_i}(x)$ function is depicted in Figure 3.

Then we have

$$P(j \text{ can be served after } i) = P(X_i + t_{ij} \le Y_j).$$

Two special cases can easily be handled. If $ef_i + t_{ij} > lp_j$, then j cannot be served after i by the same vehicle, hence, $P(j \text{ can be served after } i) = 0$. On the other hand, if $lf_i + t_{ij} \le ep_j$ then $P(j \text{ can be served after } i) = 1$. Hence, in the sequel we assume that

$$ef_i + t_{ij} < lp_j,$$
$$lf_i + t_{ij} > ep_j.$$

In general, we have

$$P(X_i + t_{ij} \le Y_j) = \int_{ef_i}^{lf_i} f_{X_i}(x) P(Y_j \ge x + t_{ij}) dx. \tag{3}$$

Since f_{X_i} is piecewise linear, it is not easy to compute (3). In the appendix, we provide closed form expressions. However, we can approximate this probability by setting $w = 0$. In this case X_i is uniformly distributed on $[ef_i, lf_i]$, which is assumed in the rest of this section.

Let $p := P(X_i \le lp_j - t_{ij})$, $\tilde{p} := P(X_i \le ep_j - t_{ij})$, $q := P(Y_j \ge ef_i + t_{ij})$, and $\tilde{q} := P(Y_j \ge lf_i + t_{ij})$. Then, we distinguish four cases, see Figure 4.

$$P(X_i + t_{ij} \le Y_j) = \begin{cases} pq/2 & \text{if } p < 1, q < 1 \\ (q + \tilde{q})/2 & \text{if } p = 1, q < 1 \\ (p + \tilde{p})/2 & \text{if } p < 1, q = 1 \\ 1 - (1 - \tilde{p})(1 - \tilde{q})/2 & \text{if } p = 1, q = 1. \end{cases} \tag{4}$$

The formulas in the respective cases correspond to the dotted areas in Figure 4. Notice that in the figure, p, q, $\tilde{p}$, and $\tilde{q}$ indicate *ratios* of the length of line segments to the length of the corresponding sides of the rectangles.

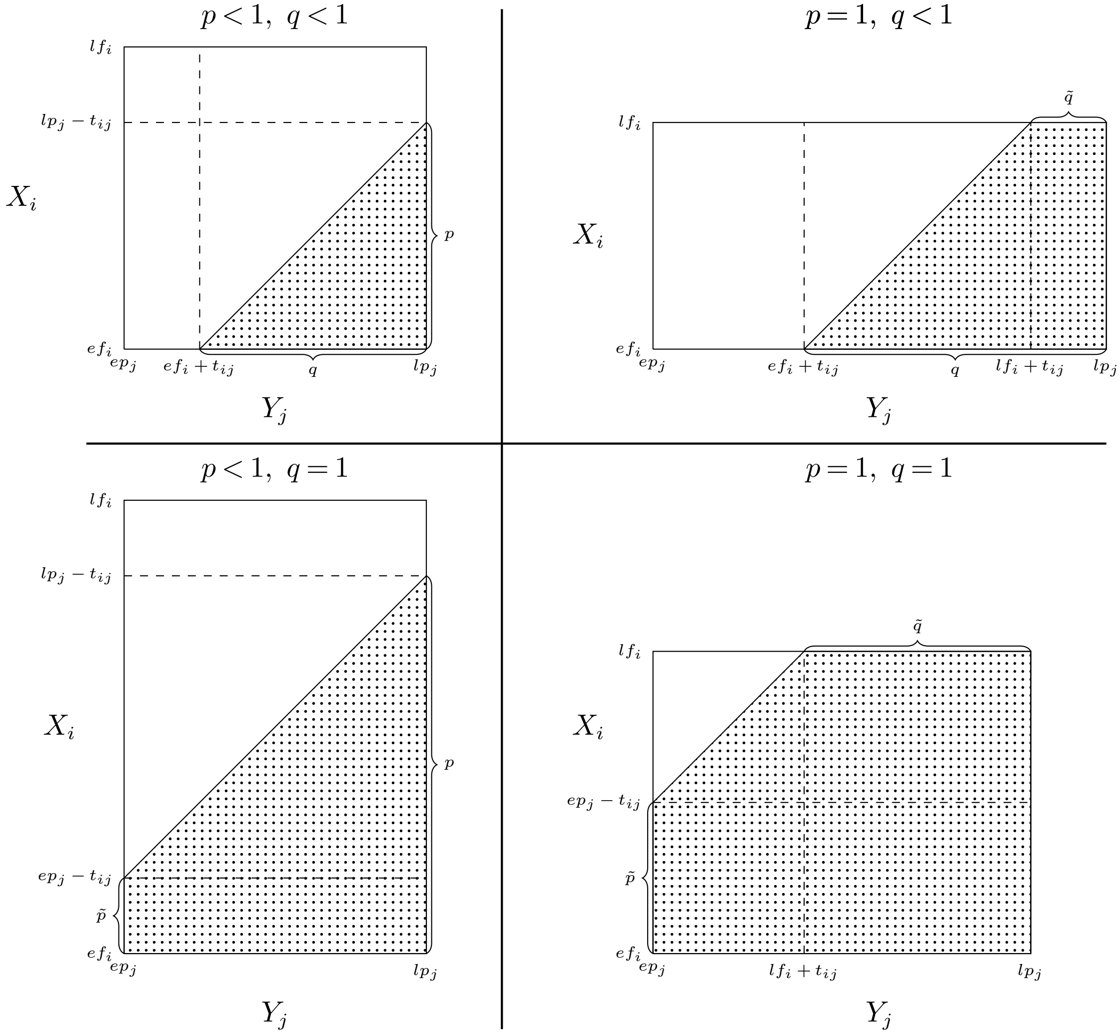


Figure 4 **Probability of $X_i + t_{ij} \leq Y_j$. Dotted areas represent the values of (X_i, Y_j) with $X_i + t_{ij} \leq Y_j$.**

4.3. Partial execution of commands

In this section we describe a simple technique to reduce the total distance traveled idle of the vehicles.

Suppose a vehicle gets a command to go to a customer which has not confirmed its request yet. This could be a good idea, because if the announcement lead times are short, and the time windows are narrow, when a customer i, say, announces its time window $[e_i, \ell_i]$ at time c_i and there is no vehicle nearby which could arrive to the pickup location of i before ℓ_i, then the customer has to be rejected, and it is penalized in the objective function (1).

On the other hand, suppose a vehicle v is on the way to some customer i, and upon arriving the the pickup location of i, another customer, say j, not too far away confirms its request, then v may go to the other customer to serve it, and later some another vehicle v_2 may serve i. However, this can be a D-tour for v. To reduce the total deadhead costs, the vehicles can apply the following

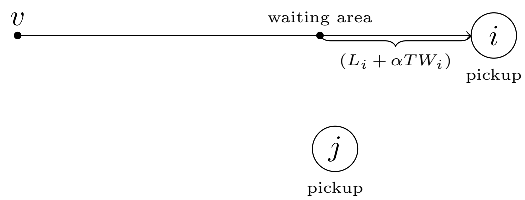


Figure 5 Partially approaching the pickup location of a customer

strategy. Instead of going to the pickup location of i, the designated vehicle v only approaches i at a distance such that the time needed to arrive to the pickup location of i is $L_i + \alpha TW_i$. This guarantees that when i confirms its request at time c_i, then at time $c_i + L_i + \alpha TW_i$, vehicle v can arrive to the pickup location of i. Since $L_i = e_i - c_i$ by definition, this means that v can arrive to i after an α fraction of the desired time window of i has passed. We call this strategy *partial execution with parameter* α. On the other hand, if the vehicles always go to the pickup location of the unconfirmed requests, then they follow the *full execution* strategy.

For an illustration, see Figure 5. In the figure, we assume that vehicle v has a unit speed, so time is equivalent to distance traveled. Since the travel time from the pickup location of i to the pickup location of j is larger than that from the waiting area to j, should j confirm its request before i, the service provider could modify the routing of v at a smaller cost.

In the next section we will demonstrate that this simple strategy can reduce the total deadhead cost.

5. Computational results

In this section we give some details of the computer implementation of our solver, describe briefly the test data, and then summarize our results. Our test data is from Srour, Agatz, and Oppen (2016), and in discussing our results, we closely follow their presentation to get a fair comparison.

5.1. Implementation

To assess the performance of our method, we have implemented a simple simulation environment in C++. For solving the minimum cost flow problems, we have used Google Optimization Tools of Google Inc. (2016). The simulation runs were very fast, the entire run took only a fraction of a second on a modern notebook computer, so computational times are not provided. The large computation speed is due to the efficient solvability of minimum cost flow problems, see e.g., Ahuja, Magnanti, and Orlin (1993). Further on, at each decision point, a single run of the minimum cost flow algorithm suffices. For computing the arc costs, we have used the formula (4) instead of the more involved ones described in the appendix. The threshold value for the probability of picking an arc has been set to 0.01. By default, we run our method with $\alpha = 0$, i.e., the vehicles approach the pickup location of unconfirmed customers to a distance of L_i. We call this the *baseline implementation*.

We have mentioned in Section 4 that the structure of the network permits cycles in the solution (consisting of arcs with unit flow) containing only customer nodes. We have developed a variant of our baseline implementation in which if a cycle is detected in a solution, then we eliminate one arc from each strongly connected component of the directed graph consisting of the arcs with positive flow values. The method was still very fast, in 2-3 iterations we got a solution without any cycles, but this extra work had insignificant impact on the entire simulation runs. We have also investigated the effect of computing the probability (3) exactly using the formulas in the appendix, but again, this had insignificant impact on the average results that we summarize in the next sections.

Unless otherwise stated, we used the baseline implementation in obtaining the results of the following sections.

5.2. Test data

We compared the results of our algorithm to those of Srour, Agatz, and Oppen (2016) on their input files (https://sites.google.com/site/pdptwinstances/, accessed 31 March 2017). These test cases are based on transport data from a dial-a-chauffeur service in The Netherlands. The parameters that determine the total cost of the service are the same in each case: $f = 6$, $g = 2.7$ and $h = 0.3$ (cf. Section 3). There are 9 vehicles and 20 customers in each instance, hence, the networks to be handled have at most 51 nodes and a few hundred of arcs. The pickup and the drop-off locations are in a 100 square kilometer area and the depot is located at a corner of this area. The vehicles can travel in a straight line between any two points at unit (1 km/min) speed. This latter assumption means that travel time (in minutes) and distance traveled (in kilometers) have the same nominal values.

The test data contains instances with different geographies, announcement lead times, time windows and parameters Δ (recall that e_i is uniformly distributed in $[\hat{e}_i - \Delta, \hat{e}_i + \Delta]$). The preannounced pickup times, $\hat{e}_i$, are drawn from a uniform distribution spanning a 6 hour period of operation, while the confirmation times are determined by earliest pickup time e_i and by the announcement lead time L_i, i.e. $c_i + L_i = e_i$. The preannounced time windows are known from the beginning.

The default setting is the following: each announcement lead time as well as the length of the time windows is 5 minutes, $L_i = TW_i = 5$ for each $i \in J$, while the value of Δ is 60. The geography of the customers is based on the concept of a center region like a city center: there are 4 customers who would like to go from an outlying region to the center, 6 customers who would like to go out from the center and the last 10 customers have random pickup and drop-off locations (geography BUS).

Srour et al. developed several test cases where they varied in most cases only one of the parameters while keeping the others at the default values. Notice that in any instance, all customers have the same L_i, and TW_i values, respectively. There are test cases with modified announcement lead times (0, 15, 30 and 60), modified time window lengths (10, 15, 30 and 60), modified Δ values (30, 45, 90 and 120), and modified geographies (IO20, where each customer wants to go out from the center and RR20, where each customer has random pickup and drop-off location). For each setting they generated 100 different test cases.

Figure 6 depicts a sample run of our method on a single problem instance. Diamonds indicate the pickup locations (labeled with j<id>s), and circles the drop-off locations (labeled with j<id>e), where <id> is the job identifier between 0 and $|J|-1$. The lines indicate the vehicle tours, and they are distinguished by colors in the electronic version.

5.3. Results with varying Δ

In this set of experiments, we consider datasets with varying Δ values (100 instace for each Δ), and with $L_i = TW_i = 5$, and geography = BUS (the default values). In Table 1, we compare our results to those of Srour, Agatz, and Oppen (2016). The table is divided into 6 sections. The first line is obtained by using complete information, i.e., using the time windows $[e_i, \ell_i]$, and solving the entire, static problem by a MIP solver. Then there are 5 additional sections corresponding to the results with the given Δ values. The rows MTS-seq depict the best results of Srour, Agatz, and Oppen (2016), and the rows 'our' indicate the corresponding results obtained by our method. In the first three columns, the average percent deviation to the complete information case is provided, while in the last four columns the average rejection costs, the average number of rejections, the number of instances without any rejections, and the average deadhead distance (over served jobs) is given. Observe that our method constantly provides significantly better results in all aspect, except the minimum cost, and the deadhead costs, where we are sometimes slightly worse). Of course, for any delta, either method is worse than the perfect information case, but for instance the average deadhead distance is close to the optimum for both methods.

5.4. Results with varying L_i

In this section results with varying the announcement lead times L_i are compared to those of Srour et al, see Figure 7(a). The results are obtained using the datasets (100 instance for each L_i) with $TW_i = 5$, $\Delta = 60$, and geography = BUS (the default values). In the figure the baseline is obtained by using perfect information, and we compare the performance of MTS-seq of Srour et al. and our method. Notice that when the announcement lead time is short, i.e., the service provider gets the confirmed time window only shortly before the service may start, our method performs significantly better, and our advantage decreases for the large lead times 30 and 60.

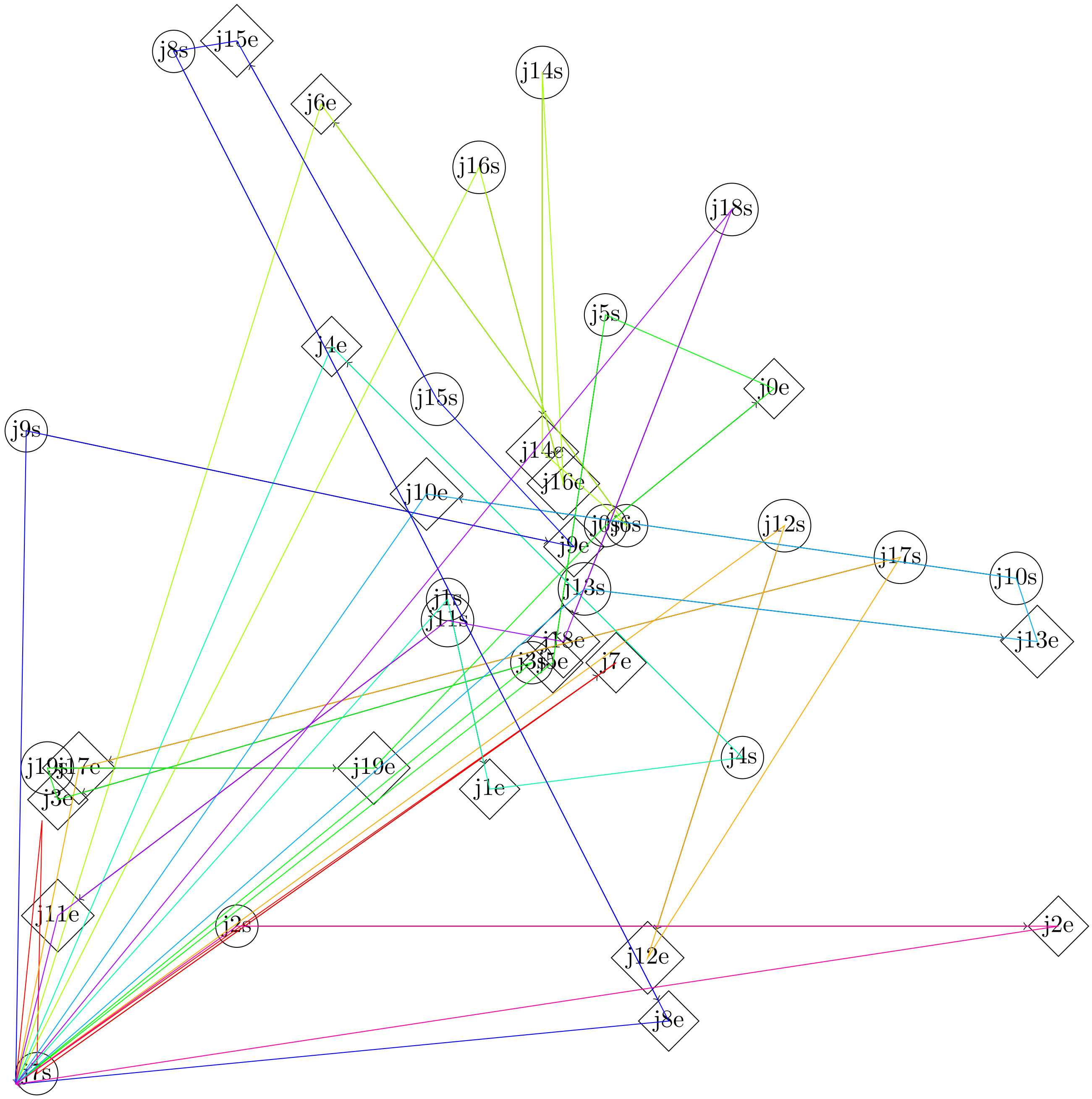


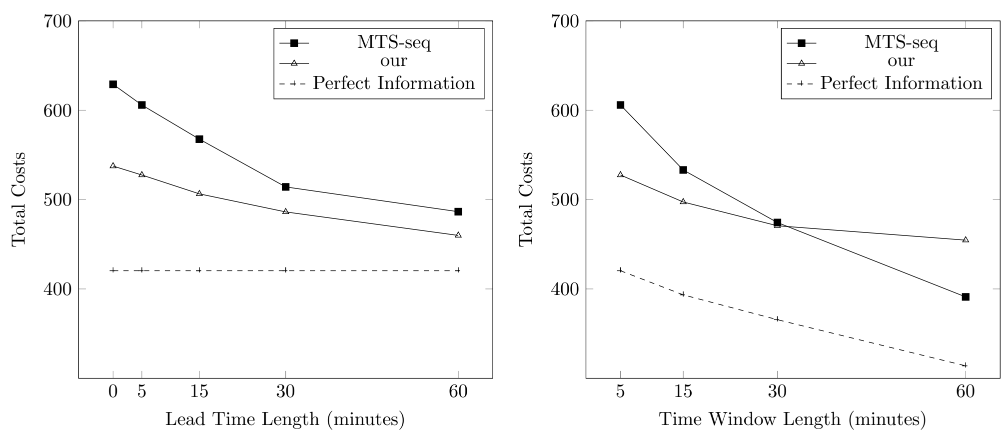
Figure 6 **Sample run of the method**

5.5. Results with varying TW_i

In Figure 7(b), the impact of varying the length of the time windows TW_i on the performance of various methods is depicted. Clearly, the perfect information case also benefits from larger time windows, so its cost curve decreases as the length of the time windows increase. Our method strongly dominates MTS-seq for short time windows, which we believe is harder to handle, it has similar performance for $TW_i = 30$, and it gives worse results for long time windows ($TW_i = 60$). Notice that in this case, our method with greater α value achieves better results, see Section 5.8.

Table 1 Impact of varying Δ, averages are taken over 100 instances

	% diff. of Perfect Info.						
	Avg. Cost	Min. Cost	Max. Cost	Avg. Rejection costs	Avg. num. of Rejections	Num. Inst. with no Rejections	Empty Dist. per Job Served
Perfect information	0.0	0.0	0.0	13	0.2	82	68.8
Range60($\Delta = 30$)							
MTS-seq	24.0	1.2	102.3	77.2	0.7	42	77.0
our	14.5	0.1	50.0	33.6	0.5	61	76.8
Range90($\Delta = 45$)							
MTS-seq	32.9	0.0	99.5	109.2	1.0	36	79.0
our	20.7	1.2	60.8	48.5	0.8	48	79.6
Range120($\Delta = 60$)							
MTS-seq	44.0	2.3	136.9	158.5	1.4	25	80.4
our	25.5	4.3	65.6	64.2	0.9	41	81.1
Range180($\Delta = 90$)							
MTS-seq	60.5	7.8	183.8	226.8	1.9	9	82.7
our	38.8	7.4	94.3	122.6	1.6	19	83.7
Range240($\Delta = 120$)							
MTS-seq	88.0	10.1	221.0	349.5	2.8	1	85.9
our	47.1	11.1	114.1	162.7	2.0	8	84.7



(a) Lead Time (b) Time Window Impact

Figure 7 Comparison of the methods for varying announcement lead times, and time window length

Since with large time windows, it is easier to serve all the clients, the exact solution of the routing problem in MTS-seq is a better strategy than ours in this case.

5.6. Results with varying Δ and TW_i

Like Srour et al, we have also made experiments with varying Δ and TW_i parameters. In Figure 8 we compare our method to MTS-seq on 2×5 datasets, i.e., one series with datasets such that $TW_i = 5$ and $\Delta \in \{30, 45, 60, 90, 120\}$ (solid lines), and another with $TW_i = 30$ and Δ from the same set (dashed lines). Notice that the range in the figure is just $2 \times \Delta$, and our figure has similar content to Figure 7 of Srour, Agatz, and Oppen (2016). Also note that the results with $TW_i = 5$ are already summarized in Table 1, although in the table we compare the performance of the methods to the perfect information case. Observe than on both series of datasets, our algorithm provides superior results to MTS-seq, and in fact as the range (Δ) increases, the difference between the performance of the two methods increases as well.

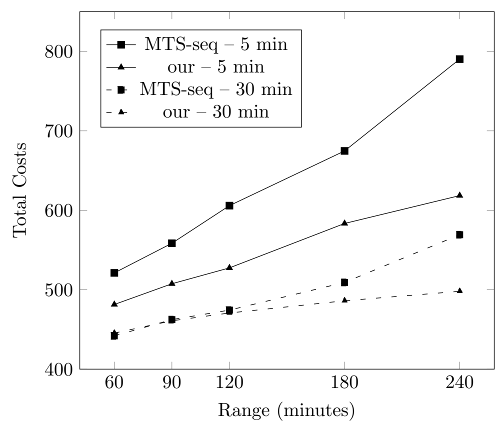


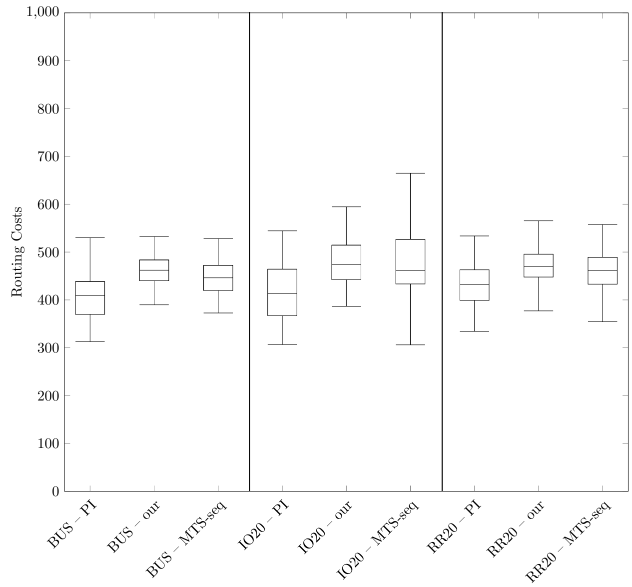
Figure 8 **Results with varying time window and range**

5.7. Results with varying geography

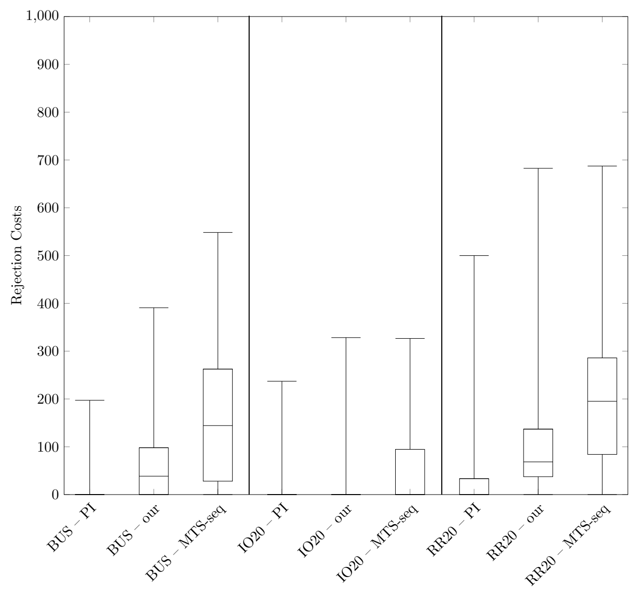
Now $L_i = TW_i = 5$, $\Delta = 60$ (the default values), but the geography of the pickup and drop-off locations is varied. Figure 9(a) shows the routing costs of the different methods on instances with the different geographies (100 instance for each geography). The method PI stands for the perfect information case (solved by a MIP solver), and *our* refers to our method and MTS-seq is that of Srour et al. In each case the figure depicts the routing cost of the instance with the lowest, the 25th, the 50th, the 75th and highest (100th) routing cost, thus we can see roughly the distribution of the routing costs. E.g. the first column shows that there are 25 BUS instances, where the routing cost is between 313 and 370 in case of complete information, 25 other, where it is between 370 and 409, etc. In Figure 9(b), we can see the same in case of the rejection costs. Observe that while the routing costs of the solution found by our method and MTS-seq are similar, our method produces significantly lower rejection costs than MTS-seq for each geography.

5.8. The impact of partial execution

In this section we summarize the results of the method described in Section 4.3. Since the method modifies the algorithm significantly only if the announcement lead times or the time windows of the instance are relatively long, thus Table 2 depicts the average costs on two 100-element datasets, one with $L_i = 60$, where the other parameters are at default values, and another with $TW_i = 60$,



(a) Routing Costs



(b) Rejection Costs

Figure 9 Minimum, 1st quartile, median, 3rd quartile, and maximum routing costs (a), and rejection costs (b) for the perfect information case, and for our and MTS-seq methods, respectively

while the other parameters are at default values. The first 3 columns depict results obtained by our method with full and partial execution strategies (see Section 4.3), while the last column depicts the reference data of MTS-seq. On the dataset with $L_i = 60$, full execution provides slightly weaker results than MTS-seq, and partial execution with $\alpha = 0$ or $\alpha = 0.3$ are both better than MTS-seq. On the dataset with $TW_i = 60$ our method provides the best results with $\alpha = 0.9$, which is still weaker than MTS-seq. This is the only dataset where our method is not competitive with MTS-seq.

Table 2 Results with full and partial execution strategies

	our method				MTS-seq
	full execution	$\alpha = 0$	$\alpha = 0.3$	$\alpha = 0.9$	
$L_i = 60$	487.4	459.9	460.5	–	486.5
$TW_i = 60$	458.2	454.6	438.3	419.7	391.1

6. Conclusions

In this paper we have studied a stochastic pickup and delivery problem proposed recently by Srour, Agatz, and Oppen (2016). We demonstrate that a simple algorithm may outperform a more heavy scenario-based approaches on several classes of problem instances. Our findings open up a number of further directions. For instance, for the specific problem, can we make better routing decisions in order to improve the results when large time windows allow more room for optimization? Can a similarly simple approach be effective in other dynamic and stochastic vehicle routing problem?

Acknowledgments

The authors are indebted to F. Jordan Srour for providing a lot of help including tabular data from her paper. This work has been supported by the OTKA grant K112881, and by the GINOP-2.3.2-15-2016-00002 grant of the Ministry of National Economy of Hungary.

Appendix A: Closed-form expressions for (3)

Below we provide some details of computing (3) when X_i is distributed according to the piecewise linear function (2) with $w > 0$. Notice that if $x \geq lp_j - t_{ij}$, then $P(Y_j \geq x + t_{ij}) = 0$, and if $Y_j < ef_i + t_{ij}$ then no $x \geq ef_i$ exists with $Y_j \geq x + t_{ij}$. So, in order to obtain a closed-form expression for (3) in the form of some integrals, we will use conditional probabilities. That is, let $p := P(X_i \leq lp_j - t_{ij})$ and $q := P(Y_j \geq ef_i + t_{ij})$. First suppose that $p < 1$ and $q < 1$. Observe that $p < 1$ implies $lf_i > lp_j - t_{ij}$, and $q < 1$ implies $ep_j < ef_i + t_{ij}$. Then

$$\begin{aligned} P(X_i + t_{ij} \leq Y_j) &= pqP(X_i + t_{ij} \leq Y_j \mid X_i \leq lp_j - t_{ij} \ \& \ Y_j \geq ef_i + t_{ij}) \\ &= pq \int_{ef_i}^{lp_j - t_{ij}} f_{X_i | X_i \leq lp_j - t_{ij}}(x) P(Y_j \geq x + t_{ij} | Y_j \geq ef_i + t_{ij}) dx \end{aligned}$$

where $f_{X_i | X_i \leq lp_j - t_{ij}}(x)$ is the conditional probability density function defined as

$$f_{X_i | X_i \leq lp_j - t_{ij}}(x) = \left(\int_{ef_i}^{lp_j - t_{ij}} f_{X_i}(z) dz \right)^{-1} f_{X_i}(x).$$

Then we have

$$P(X_i + t_{ij} \leq Y_j) = pq \int_{ef_i}^{lp_j - t_{ij}} f_{X_i | X_i \leq lp_j - t_{ij}}(x) \left(\int_{x + t_{ij}}^{lp_j} f_{Y_j | Y_j \geq ef_i + t_{ij}}(y) dy \right) dx, \tag{5}$$

where $f_{Y_j | Y_j \geq ef_i + t_{ij}}(y)$ is the conditional probability density function

$$f_{Y_j | Y_j \geq ef_i + t_{ij}}(y) = \left(\int_{ef_i + t_{ij}}^{lp_j} f_{Y_j}(z) dz \right)^{-1} f_{Y_j}(y) = \frac{1}{lp_j - ef_i - t_{ij}}.$$

Now (5) can be rewritten as

$$pq \int_{ef_i}^{lp_j - t_{ij}} f_{X_i|X_i \le lp_j - t_{ij}}(x) \left(\frac{lp_j - x - t_{ij}}{lp_j - ef_i - t_{ij}} \right) dx, \tag{6}$$

Since $f_{X_i|X_i \le lp_j - t_{ij}}$ is piecewise linear, (6) can be expressed as the sum of three integrals:

$$\begin{aligned} pq \left(\int_{ef_i}^{lp_j - t_{ij}} f_{X_i}(z) dz \right)^{-1} & \left(\int_{ef_i}^{\min\{ef_i + w, lp_j - t_{ij}\}} \left(\frac{x - ef_i}{w(lf_i - ef_i - w)} \right) \left(\frac{lp_j - x - t_{ij}}{lp_j - ef_i - t_{ij}} \right) dx \right. \\ & + \int_{ef_i + w}^{\min\{lf_i - w, lp_j - t_{ij}\}} \left(\frac{1}{(lf_i - ef_i - w)} \right) \left(\frac{lp_j - x - t_{ij}}{lp_j - ef_i - t_{ij}} \right) dx \\ & \left. + \int_{lp_j - w}^{lp_j - t_{ij}} \left(\frac{lf_i - x}{w(lf_i - ef_i - w)} \right) \left(\frac{lp_j - x - t_{ij}}{lp_j - ef_i - t_{ij}} \right) dx \right). \end{aligned}$$

Notice that we define $\int_a^b g(z)dz = 0$ if $a \ge b$.

Now suppose that $p < 1$ and $q = 1$, i.e., $P(Y_j \ge ef_i + t_{ij}) = 1$. Then $ep_j \ge ef_i + t_{ij}$, and for $x \in [ef_i, ep_j - t_{ij}]$ we have $P(Y_j \ge x + t_{ij}) = 1$. Consequently,

$$\begin{aligned} P(X_i + t_{ij} \le Y_j) &= p \int_{ef_i}^{lp_j - t_{ij}} f_{X_i|X_i \le lp_j - t_{ij}}(x) P(Y \ge x + t_{ij}) dx \\ &= p \left(\int_{ef_i}^{ep_j - t_{ij}} f_{X_i|X_i \le lp_j - t_{ij}}(x) dx + \int_{ep_j - t_{ij}}^{lp_j - t_{ij}} f_{X_i|X_i \le lp_j - t_{ij}}(x) P(Y \ge x + t_{ij}) dx \right) \\ &= p \left(\int_{ef_i}^{ep_j - t_{ij}} f_{X_i|X_i \le lp_j - t_{ij}}(x) dx + \int_{ep_j - t_{ij}}^{lp_j - t_{ij}} f_{X_i|X_i \le lp_j - t_{ij}}(x) \left(\frac{lp_j - x - t_{ij}}{lp_j - ep_j} \right) dx \right). \end{aligned} \tag{7}$$

Using the piecewise linearity of $f_{X_i|X_i \le lp_j - t_{ij}}(x)$, the two integrals in (7) can be decomposed into the sum of 3 elementary integrals each, the details are omitted.

The third case arises when $q < 1$ and $p = 1$, i.e., $P(X_i \le lp_j - t_{ij}) = 1$. Then $lp_j - t_{ij} \ge lf_i$. Hence,

$$\begin{aligned} P(X_i + t_{ij} \le Y_j) &= q \int_{ef_i}^{lf_i} f_{X_i}(x) P(Y \ge x + t_{ij} | Y_j \ge ef_i + t_{ij}) dx \\ &= q \left(\int_{ef_i}^{ep_j - t_{ij}} f_{X_i}(x) dx + \int_{ep_j - t_{ij}}^{lf_i} f_{X_i}(x) \left(\frac{lp_j - x - t_{ij}}{lp_j - ef_i - t_{ij}} \right) dx \right). \end{aligned} \tag{8}$$

Finally, suppose $p = 1$ and $q = 1$. Then we have $lp_j - t_{ij} \ge lf_i$ and $ep_j \ge ef_i + t_{ij}$. Consequently, we have

$$\begin{aligned} P(X_i + t_{ij} \le Y_j) &= \int_{ef_i}^{lf_i} f_{X_i}(x) P(Y \ge x + t_{ij}) dx \\ &= \int_{ef_i}^{ep_j - t_{ij}} f_{X_i}(x) dx + \int_{ep_j - t_{ij}}^{lf_i} f_{X_i}(x) \left(\frac{lp_j - x - t_{ij}}{lp_j - ep_j} \right) dx. \end{aligned} \tag{9}$$

References

Ahuja RK, Magnanti TL, Orlin JB, 1993 *Network flows: theory, algorithms, and applications* (Prentice hall).

Albareda-Sambola M, Fernández E, Laporte G, 2014 *The dynamic multiperiod vehicle routing problem with probabilistic information. Computers & Operations Research* 48:31–39.

Bent RW, Van Hentenryck P, 2004 *Scenario-based planning for partially dynamic vehicle routing with stochastic customers. Operations Research* 52(6):977–987.

Berbeglia G, Cordeau JF, Laporte G, 2010 *Dynamic pickup and delivery problems. European Journal of Operational Research* 202(1):8–15.

Bertsimas DJ, Van Ryzin G, 1991 *A stochastic and dynamic vehicle routing problem in the euclidean plane. Operations Research* 39(4):601–615.

Bertsimas DJ, Van Ryzin G, 1993 *Stochastic and dynamic vehicle routing in the euclidean plane with multiple capacitated vehicles. Operations Research* 41(1):60–76.

Branke J, Middendorf M, Noeth G, Dessouky M, 2005 *Waiting strategies for dynamic vehicle routing. Transportation science* 39(3):298–312.

Ferrucci F, Bock S, Gendreau M, 2013 *A pro-active real-time control approach for dynamic vehicle routing problems dealing with the delivery of urgent goods. European Journal of Operational Research* 225(1):130–141.

Gendreau M, Guertin F, Potvin JY, Taillard E, 1999 *Parallel tabu search for real-time vehicle routing and dispatching. Transportation science* 33(4):381–390.

Google Inc, 2016 *Google optimization tools.* URL `https://developers.google.com/optimization/`.

Günlük O, Kimbrel T, Ladanyi L, Schieber B, Sorkin GB, 2006 *Vehicle routing and staffing for sedan service. Transportation Science* 40(3):313–326.

Ho SC, Haugland D, 2011 *Local search heuristics for the probabilistic dial-a-ride problem. Or Spectrum* 33(4):961–988.

Hvattum LM, Løkketangen A, Laporte G, 2006 *Solving a dynamic and stochastic vehicle routing problem with a sample scenario hedging heuristic. Transportation Science* 40(4):421–438.

Ichoua S, Gendreau M, Potvin JY, 2006 *Exploiting knowledge about future demands for real-time vehicle dispatching. Transportation Science* 40(2):211–225.

Mitrović-Minić S, Krishnamurti R, Laporte G, 2004 *Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. Transportation Research Part B: Methodological* 38(8):669–685.

Mitrović-Minić S, Laporte G, 2004 *Waiting strategies for the dynamic pickup and delivery problem with time windows. Transportation Research Part B: Methodological* 38(7):635–655.

Muñoz-Carpintero D, Sáez D, Cortés CE, Núñez A, 2015 *A methodology based on evolutionary algorithms to solve a dynamic pickup and delivery problem under a hybrid predictive control approach. Transportation Science* 49(2):239–253.

Pillac V, Gendreau M, Guéret C, Medaglia AL, 2013 *A review of dynamic vehicle routing problems. European Journal of Operational Research* 225(1):1–11.

Psaraftis HN, 1988 *Dynamic vehicle routing problems.* Golden B, Assad A, eds., *Vehicle Routing: Methods and Studies*, volume 16, 223–248.

Psaraftis HN, Wen M, Kontovas CA, 2016 *Dynamic vehicle routing problems: Three decades and counting. Networks* 67(1):3–31.

Srour FJ, Agatz N, Oppen J, 2016 *Strategies for handling temporal uncertainty in pickup and delivery problems with time windows. Transportation Science* URL `http://dx.doi.org/10.1287/trsc.2015.0658`.

Thomas BW, 2007 *Waiting strategies for anticipating service requests from known customer locations. Transportation Science* 41(3):319–331.

Thomas BW, White III CC, 2004 *Anticipatory route selection. Transportation Science* 38(4):473–487.

Vonolfen S, Affenzeller M, 2016 *Distribution of waiting time for dynamic pickup and delivery problems. Annals of Operations Research* 236(2):359–382.

Yang J, Jaillet P, Mahmassani H, 2004 *Real-time multivehicle truckload pickup and delivery problems. Transportation Science* 38(2):135–148.