

# Strategies for Handling Temporal Uncertainty in Pickup and Delivery Problems with Time Windows

F. Jordan Srour      Niels Agatz      Johan Oppen

August 12, 2015

## Abstract

In many real-life routing problems there is more uncertainty with respect to the required timing of the service than with respect to the service locations. We focus on a pickup and delivery problem with time windows in which the pickup and drop-off locations of the service requests are fully known in advance, but the time at which these jobs will require service is only fully revealed during operations. We develop a sample-scenario routing strategy to accommodate a variety of potential time realizations while designing and updating the routes. Our experiments on a breadth of instances show that advance time related information, if used intelligently, can yield benefits. Furthermore, we show that it is beneficial to tailor the consensus function that is used in the sample-scenario approach to the specifics of the problem setting. By doing so, our strategy performs well on instances with both short time windows and limited advance confirmation.

## 1 Introduction

Most vehicle routing problems encountered in practice, suffer from an initial level of uncertainty regarding some of the relevant parameters. In the context of truckload, vehicle routing problems, this uncertainty in the relevant parameters can be classified along two dimensions

– spatial and temporal. Spatial uncertainty refers to situations in which the exact location of a job is not known (or known inaccurately) during planning, but becomes known at some point in operations. Temporal uncertainty, on the other hand, refers to situations in which the exact service time or time window of a job is not known (or known inaccurately) until operations are underway. These two dimensions are so intertwined that focusing on one of them separately is typically difficult.

Regarding truckload, pickup and delivery problems, the literature tends to focus on scenarios in which both the release time and the location of future jobs are unknown. For such problems, multiple solution techniques exist ranging from reactive, online/dynamic, rolling horizon strategies to stochastic strategies (Berbeglia et al. 2010). The isolated study of temporal uncertainty is, however, less present in the literature. We seek to fill this gap by studying strategies for coping with time window uncertainty given a set of known service locations across a collection of pickup and delivery problems with time windows.

Despite its absence from the literature, this type of uncertainty is common in practice. For example, harbor pilots, who must bring ships to berth, know where they will meet the ship and where it will berth, but the exact arrival time of the ship is often unknown. In container transport, a drayage provider often knows the location of the terminals for pickup and drop-off, but does not know when the container will be released from the pickup terminal. This problem is also common in passenger transportation when a patient needs to be driven home following a medical procedure, but the duration of the procedure is uncertain. A recently growing business area (Bandyk 2012), that also faces this uncertainty, is on-demand chauffeur services that transport customers home in their own vehicle as a safe alternative to driving home inebriated.

In all of these contexts, it is common for the customers to alert the service provider to the need for transport without initially committing to a specific time. In such preannouncements, the customer will quote both the pickup and drop-off location along with an approximate pickup time, e.g. around 11am. Later, during operations, when the preannounced customer actually requires service, they call the company for pickup. In return for this advance notification, the customer expects to receive fast service when they call in. If announced early

enough, the location information provided by the preannouncements may facilitate more efficient, operational route plans. However, it is unclear how to incorporate the probabilistic time information into the route planning in a way that maximizes the operational benefits.

To the best of our knowledge, this is the first paper to study the impact of uncertain advance time window information in vehicle routing. Specifically, our contributions are the introduction and explanation of a relevant new planning problem – the pickup and delivery problem with confirmed locations and uncertain time windows; the development of a novel sample-scenario routing strategy to deal with uncertain advance time window information; and a demonstration of the efficacy of our strategy through experiments based on data inspired by a real-world application. Our results show that having information on the service locations can be beneficial even if there is significant uncertainty about the timing.

In the next section, we discuss the relevant literature. Section 3 describes our formulation of the problem and the off-line routing model. In Section 4, we introduce the dynamic routing strategies that we test. In Section 5, we describe the construction and structure of the problem instances used to test the strategies. In Section 6, we present the results of the computational experiments. We conclude the paper in Section 7 with a discussion of our findings and future research directions.

## 2 Related Literature

The problem that is discussed in this paper can be categorized as a stochastic and dynamic pickup and delivery problem. The problem is stochastic because the information about the timing of the jobs can be described by a random variable with a known distribution. The problem is dynamic because the available information evolves over time – with accuracy at its worst during initial planning and at its best during operations. What is particularly interesting in our context is that the job locations are known in advance while their timing is uncertain.

The literature provides several approaches to cope with this evolution of information: through the incorporation of stochastic information during planning (Dror et al. 1989),

through update methods, where a static problem is solved repeatedly (Berbeglia et al. 2010, Pillac et al. 2012a), through online algorithms where a specific action is taken in response to new information arriving (Jaillet and Wagner 2006), or through stochastic, dynamic strategies that combine the first two approaches (Goodson et al. 2013). When incorporating stochastic information in the planning phase, the goal is to generate a robust plan – one that remains relevant despite changes in information. Both dynamic and online algorithms are similar in that new routes or routing actions are only undertaken when the uncertain information becomes certain. We combine a stochastic strategy with a dynamic strategy by merging a set of stochastic time related information with a set of certain information, for both location and time windows, to create and update plans during operations.

The issue of exploiting knowledge on future demand information has received a lot of attention in the last decade (see Ichoua et al. (2006) for an overview). This line of research typically shows that explicitly taking probabilistic spatial information into account is beneficial in terms of the solution quality. Even after the vehicles are in transit, stochastic, dynamic strategies that accommodate job location information on unknown future jobs demonstrate improvements (see for example Thomas and White (2004), Hyytiä et al. (2012), Cortes et al. (2009)). Others use probabilistic spatial information to address the issue of choosing where the vehicles should idle in anticipation of future requests (Larsen et al. 2004, Thomas 2007). Still others use probabilistic or advance information to make decisions about which jobs to serve (Jaillet and Lu 2011, Kim et al. 2004).

Considering the problem from the perspective of decision making, Approximate Dynamic Programming (ADP) provides a tool to decompose the problem into a series of decisions over time. The key to using ADP effectively is to have a clear relationship on the transition from one system state to the next both pre- and post-decision and to have a mechanism by which to evaluate the different decision policies. Powell et al. (1988) utilized ADP to study a problem of truckload pickup and delivery when the job requests are uncertain. More recently, Goodson et al. (2013) used a roll-out algorithm along the lines of ADP in order to serve loads when the size of the loads was uncertain. Our problem contrasts to both of these settings in that we know both the job locations and load size with certainty – it is the time

that the jobs will be released that is uncertain. As such, the main challenge is defining the proper state transitions in a way that is tractable. The solution strategy that we choose to overcome this challenge is that of scenario sampling.

Several papers in the area of stochastic, dynamic vehicle routing use sampling to construct a set of potential routes from which the best route is chosen at each decision epoch. Bent and Van Hentenryck (2004) use a set of potential plans that contain both the existing customers and a set of potential requests. From among these plans, they select and enact the plan with the most similarity to all of the other plans in terms of vehicle to job assignments. Hvattum et al. (2006) present a similar sampling-based strategy for a dynamic vehicle routing problem. They formulate the potential job realizations in the context of a stochastic model with recourse and solve the model repeatedly at given intervals. Ghiani et al. (2012) compare a sample-scenario based approach to a simpler anticipatory insertion heuristic in the context of a dynamic and stochastic traveling salesman problem. Pillac et al. (2012b) present a framework to generalize the strategy of Bent and Van Hentenryck (2004) across a variety of vehicle routing problems. They demonstrate the capability of this framework on the Dynamic Vehicle Routing Problem with Stochastic Demands. In our work, we use an object-oriented architecture similar to that of Pillac et al. (2012b) to implement and test our real-time strategy that uses sampling to generate a set of potential plans. However, the set of plans we generate is not based on sampling potential job locations, but sampling potential time window realizations given a set of known job locations.

Focusing on the pickup and delivery literature, as noted in Parragh et al. (2008), a multitude of articles measure the impact of real-time routing in the single vehicle, dynamic pickup and delivery problem through the use of competitive analysis. However, these results are less relevant to our work as we examine a setting with multiple vehicles. More similar to our setting, Mitrović-Minić et al. (2004) study a multi-vehicle dynamic pickup and delivery problem with time windows. They evaluate different heuristic strategies for a less-than-truckload courier service which may be “non-optimal” in the near term, but work well over a longer horizon due to the incorporation of slack. However, unlike the current work, they do not explicitly take into account any information about the locations of future requests or

their timing. Furthermore, our work differs because we focus on a setting in which each job requires the full capacity of the vehicle.

Since unit-capacity pickup and delivery problems most frequently arise in freight transportation, the problem class is termed the dynamic truckload pickup and delivery problem (Savelsbergh and Sol 1995). In the dynamic truckload pickup and delivery problem literature, the work of Yang et al. (1999, 2004) is most relevant to our work. In Yang et al. (1999), the authors introduce an efficient mixed-integer program for use in a dynamic rolling horizon strategy. They demonstrate that using a re-optimization strategy to react to the arrival of new jobs is better, in terms of routing costs, than a variety of heuristic, insertion strategies. Extending this work in Yang et al. (2004), the authors demonstrate that a rolling horizon strategy which also includes, via opportunity costs, probabilistic information on future job pickup and drop-off locations outperforms the simple reactive strategy. In contrast to the work of Yang et al. (2004), who consider all vehicles as part of one planning problem, Mes et al. (2010) introduce a dynamic pickup and delivery problem strategy for one vehicle. The pricing and scheduling strategy that they introduce is based on dynamic programming and explicitly takes into account the impact on future opportunities of each new job. The importance of including future opportunities in their strategy stems from the context they examine in which each individual vehicle is assumed to be bidding for jobs in a series of job auctions. As the realistic contexts for our problem (ship berthing, container transport, taxi services, and chauffeur services), assume a dedicated set of customers served by a fixed fleet of vehicles, the work of Yang et al. (2004) is more relevant.

### 3 The Problem Setting and Definition

We consider a stylized setting that we believe captures the essence of most truckload trucking and taxi environments. In this context, the delivery vehicles have unit capacity and can serve only one job at a time. A transportation service provider receives a sequence  $J$  of pick-up and delivery requests arriving over time.

A service request  $i \in J$  that calls in at time  $c_i$  specifies the pickup location, drop-off

location, and desired pickup time. To take into account that customers typically have some flexibility in their timing, we model the desired pickup time for request  $i$  as a service time window of length  $TW_i$  with an earliest pickup time,  $e_i$ , and a latest pickup time,  $e_i + TW_i = l_i$ . The service cannot start before the earliest pickup time and delays beyond the latest pickup time are unacceptable for the customer. We refer to the difference  $e_i - c_i$  as the announcement lead time,  $L_i$ .

A customer can also preannounce a request for service. When a job  $i \in J$  preannounces their service request at time  $a_i$ , before the actual call-in time, they state their pickup location, drop-off location, and expected pickup time window, defined by an earliest pickup time,  $\hat{e}_i$  and a latest pickup time,  $\hat{l}_i$ . As preannouncements and the uncertain time information they carry are the basis of this paper, we consider a setting in which all jobs are preannounced before the start of operations. The time line of a job and its associated information events are shown in Figure 1.

The locations are guaranteed to be accurate and all preannounced jobs will materialize, but there is uncertainty about the actual required pickup time. Nevertheless, the expected pickup time as preannounced by the customers should be the best predictor of the actual individual pickup time because it incorporates all factors that are relevant to their individual situation. However, like many predictions, these expected times cannot be taken literally and the customer may require service earlier or later than expected due to various uncontrollable external events. This is similar to the concept of forecasting error in settings where we must predict some future event. As such, the differences between the actual pickup times and the forecast pickup times are random variables known only in distribution when planning. When a certain service provider has been in operation for a while it is likely that there will be historical data available on the distribution of the error of the preannounced information. The simplest approach is to use the observed empirical distribution to predict the error distribution of a new customer (Williams and Goodman 1971). More advanced methods could involve data mining and machine learning techniques to determine the characteristics of the preannouncements that would allow for greater reliability in handling the pre-announced information.

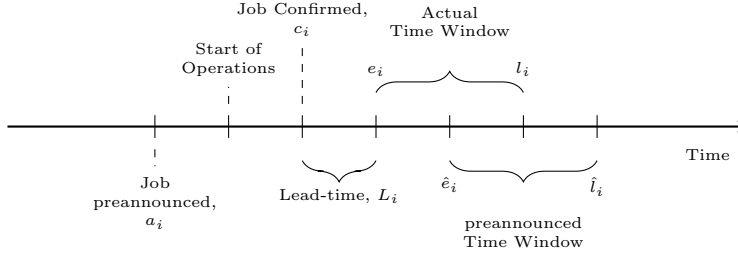


Figure 1: Time line of a preannounced job.

We assume that preannouncements arrive early enough to support decisions on how many vehicles  $|V|$  to deploy on a given day. The number of vehicles are chosen in a way that we can serve all jobs efficiently if they actually confirm for their preannounced time windows. Thus, when a job actually calls in to confirm, the available fleet of vehicles is fixed. Our objective is to find the vehicle routes that maximize the expected number of preannouncements served within their confirmed time windows while minimizing the routing costs. A job can be rejected in execution since its real time window information is only revealed at its confirmation time. Similarly, jobs that may appear infeasible initially may be serviceable once the real time window information is revealed. When a job is rejected, it means, in practice, that we have to hire an alternative (more expensive) resource to serve this job. Therefore, we penalize the rejections based on a fixed fee plus the job’s distance. In the real-world passenger transport and ship berthing applications that motivated our research, customers pre-announce their need for service upfront to ensure immediate service at the time they call-in. Therefore, we prefer to model the cost of rejections as opposed to permitting a penalized delay.

### 3.1 Formulation Off-line Model

We establish a new route plan prior to the start of operations and at each decision epoch. A decision epoch is triggered whenever a job confirms service. The route plan established at each decision epoch is, in part, found by modeling and solving a snap-shot version of the off-line routing problem. We model the off-line problem as a standard truck-load pickup and



delivery problem using a model that is very similar to the one presented in Yang et al. (1999) and Yang et al. (2004).

The model seeks a set of least cost cycles that describe the order in which each vehicle should serve the jobs, the vehicles that should remain idle, and the jobs that should be rejected. The model operates over three sets of nodes: one set of nodes,  $V$ , representing the vehicles, one set of nodes,  $J$ , representing the jobs, and one set of nodes,  $N$ , representing a combined set that includes one node for each vehicle and each job. It is important to note that each job node actually represents a revenue generating trip from a pickup location to a drop-off location. Thus, the distances between any two nodes are asymmetric.

As depicted in Figure 2, the solution resulting from this model may be a tour from Vehicle 1 to Job 1, then Job 2, then Vehicle 2, then Job 3, then back to Vehicle 1. This would indicate that Vehicle 1 serves Jobs 1 and 2, while Vehicle 2 serves Job 3. The cycle including only Vehicle 3 indicates that Vehicle 3 remains idle. Similarly, the cycle including only Job 4 indicates that Job 4 is rejected.

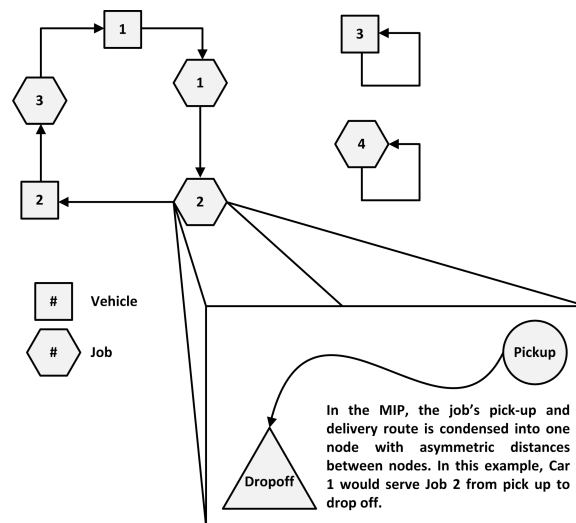


Figure 2: Cycles in the MIP solution structure.

There are two classes of decision variables: those associated with routing and those associated with the timing of the jobs. The routing variables are binary variables  $x_{ij}$  representing the presence (1) or the absence (0) of an arc between all of the nodes in both  $V$  and  $J$ . The

timing variables are the continuous variables  $\delta_i$  representing the time of arrival at the pickup location of job  $i \in J$ .

The model aims to minimize both the empty (or non-revenue generating) routing costs and the opportunity costs associated with rejecting jobs. We do not include a cost per time unit (e.g. labor costs) because we assume these to be fixed regardless of the routes. Specifically, the cost coefficients of the routing variables in the objective function are: the cost of including an arc from a node  $k \in V$  to any other node  $l \in V$ , which is the cost (0, in our case) of allowing vehicle  $k$  to remain idle;  $d_{ki}^O$  is the cost of an arc from a node  $k \in V$  to a node  $i \in J$ , which is the cost of the route from vehicle  $k$ 's current location or origin to the pickup location of job  $i$ ;  $d_{ij}$  is the cost of including an arc from a node  $i \in J$  to another node  $j \in J$ , which is the cost of traveling from the drop-off location of job  $i$  to the pickup location of job  $j$ ;  $d_{ii}$  is the cost of an arc from a node  $i \in J$  to the same node  $i \in J$ , which is the cost of rejecting job  $i$ ; and  $d_{ik}^H$  is the cost of an arc from a node  $i \in J$  to a node  $k \in V$ , which is the cost of traveling from the drop-off location of job  $i$  to the depot or home base of vehicle  $k$ . Thus, the objective function is:  $\min \sum_{k \in V} \sum_{i \in J} d_{ki}^O x_{ki} + \sum_{i \in J} \sum_{j \in J} d_{ij} x_{ij} + \sum_{i \in J} \sum_{k \in V} d_{ik}^H x_{ik}$ .

The problem is constrained by the need to include every job on a route or as part of the rejections while also obeying the time windows imposed on the pickup locations for the jobs. Mathematically, these constraints are:

$$\sum_{j \in N} x_{ij} = 1 \quad \forall i \in N \quad (1)$$

$$\sum_{i \in N} x_{ij} = 1 \quad \forall j \in N \quad (2)$$

$$\delta_i - \sum_{k \in V} (d_{ki}^O + v^k) x_{ki} \geq 0 \quad \forall i \in J \quad (3)$$

$$\delta_j - \delta_i - M x_{ij} + (d_{ii} + d_{ij}) x_{ii} \geq d_{ii} + d_{ij} - M \quad \forall i, j \in J \quad (4)$$

$$\bar{e}_i \leq \delta_i \leq \bar{l}_i \quad \forall i \in J \quad (5)$$

$$\delta_i \in \mathbb{R}^+ \quad \forall i \in J \quad (6)$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \in N \quad (7)$$

The routing related constraints, equations (1) and (2), specify that every node must have one arc that enters it and one arc that leaves it. It is possible that one arc both enters and exits the same node (i.e. a self loop) if the node is from the set  $V$ , this represents an idle vehicle; if the node is from set  $J$ , this represents a rejected job. An arc from a vehicle node to a job node is only feasible if it is time feasible as specified in equations (3) and (5), i.e. if it is possible to start service within the job’s presumed time window given the time that the vehicle becomes available,  $v^k$ , and the time required to travel from the vehicle’s available location to the pickup location. We disallow any delay in arrival to the pickup location. Similarly, as specified in equation (4) an arc from one job  $i$  to another job  $j$  is only feasible if it is time feasible to serve job  $j$  after job  $i$ . Constraints (6) and (7) specify the domain of the decision variables.

Even though we know the job locations at the start of operations, the problem is still dynamic as the job confirmations arrive over time. The following section focuses on the use of the off-line problem in a rolling horizon framework.

## 4 Real-time Strategies

We incorporate the off-line model that was presented in Subsection 3.1 in a rolling horizon framework that runs the optimization each time a job is confirmed to the system, that is at every  $c_i$ . Due to the relatively short announcement lead times there is no time to postpone the assignment decisions and buffer several requests. We assume that the optimization run is instantaneous. This is not unrealistic as the off-line optimization problem consistently solves in less than one minute on an average laptop. Figure 3 illustrates the interplay of the planning and execution events in simulation.

For ease of exposition and without loss of generality, we assume that all  $c_i$  are ordered in increasing order ( $c_i < c_{i+1}$ ). An optimization run at time  $c_i$  includes all unserved jobs that are known (confirmed or preannounced) at time  $c_i$  excluding the jobs that were rejected earlier in execution. The optimization run yields an *operational* route plan that specifies the assignment of vehicles to jobs and the time at which the assigned vehicles are scheduled to

pick up the jobs within their time windows. The operational route plan also specifies which jobs are impossible to serve within their time windows.

There are four different actions that can be undertaken following the generation of a new operational route plan: (1) move the vehicle to an assigned, preannounced job that has not confirmed yet, (2) commit the assignment of the vehicle to a confirmed job, (3) reject a confirmed job, or (4) let the vehicle wait idle at its current location, which in some cases may be at a point en route. In reality, if waiting en route is not permissible, then an additional constraint may be added to the MIP to forbid vehicles from waiting anywhere other than the depot, the pickup location or the dropoff location of a job. In our case, inspired by pickup and dropoff operations in an urban environment, waiting en route is not an issue. The specific actions taken based on the operational route plan are based on the timings of the jobs. The route plan indicates both the earliest time that a vehicle  $v$  is available to serve job  $j$ ,  $t_j^{v-}$ , and the latest time that the vehicle must depart from its current location to reach job  $j$  at the time specified in the plan,  $t_j^{v+}$ .

In an earlier study of the traveling salesman problem with confirmed locations and uncertain release times, Srour and Zuidwijk (2008) show that it is beneficial to move to a known unserved job location as soon as possible. While we are working in a different metric space with time windows rather than release times, we also adopt the policy that a vehicle, assigned to a job in the operational plan, will start moving towards it as early as possible, i.e. at time  $t_j^{v-}$ . However, the service at job  $i$  cannot start before the confirmed or actual earliest service time  $e_i$ . If a vehicle arrives at the pickup location before this time it will wait until the appropriate time. If the vehicle is moving to or waiting at the pickup location of an unconfirmed preannouncement, it may be re-routed to serve another job. We only commit to an assignment for jobs that are confirmed ( $t > c_i$ ) and only at the latest possible move time ( $t_j^{v+} = t$ ) for that job. All jobs with  $t_j^{v+} > t$  are always included in the pool of unserved jobs that are part of the optimization run at  $t$ . We reject a job if it cannot be scheduled for service within its confirmed pickup time window. As such, a preannounced job is never rejected before its confirmation time.

At the next optimization run, at time  $c_{i+1}$ , all vehicle locations are updated along with

the distance between the vehicles and all unserved jobs. If no subsequent optimization run occurs before time  $t_j^{v-}$ , then vehicle  $v$  is set to move to job  $j$  at time  $t_j^{v-}$ . Similarly, if no subsequent optimization run occurs before time  $t_j^{v+}$ , then we commit to the assignment: job  $j$  is served by vehicle  $v$ , with travel beginning at time  $t_j^{v+}$ . The commitment of vehicle  $v$  to job  $j$  is maintained in all future plans.

We term the time window information used in solving the off-line problem “the presumed time window information” and denote it as  $(\bar{e}_i, \bar{l}_i)$ . The manner by which we include the preannounced jobs with their “presumed” time window information defines our various routing strategies, the topic of the following subsections.

## 4.1 Simple Benchmark Strategies

The *Naïve* strategy represents the most straightforward way to incorporate the preannounced information. This strategy reflects what a service provider would do without sophisticated decision support tools. In this strategy, we schedule preannouncements as if the preannounced time window,  $(\hat{e}_i, \hat{l}_i)$ , is correct until the job confirms at  $c_i$ . Thus, when  $t < c_i$ , job  $i$  is included with  $(\bar{e}_i, \bar{l}_i) = (\hat{e}_i, \hat{l}_i)$  and when  $t \geq c_i$ , then job  $i$  is included with  $(\bar{e}_i, \bar{l}_i) = (e_i, l_i)$ . If a job has not confirmed at the earliest preannounced pickup time minus the announcement lead time ( $\bar{e}_i - L_i > t$ ), then the presumed earliest pickup time window is pushed further with  $(\bar{e}_i, \bar{l}_i) = (t + L_i, t + L_i + TW_i)$ .

As an even simpler benchmark, the *Ignore* strategy does not incorporate preannouncements at all until they confirm at time  $c_i$  and we know their time window. Thus, job  $i$  is not included in the optimization model until time  $t \geq c_i$  at which point  $(\bar{e}_i, \bar{l}_i) = (e_i, l_i)$ . This represents a fully dynamic, reactive strategy.

Moreover, we include an off-line benchmark, *Perfect Information*, in which all jobs are known with 100% accurate time window information.

## 4.2 The Multiple Time Scenario Strategy

We term our stochastic, dynamic strategy the *Multiple Time Scenario (MTS)* strategy as it relies on a pool of plans,  $\Pi$ , based on multiple time window scenarios for the preannouncements. At the start of operations, a set of scenarios,  $S$ , is generated. A single scenario is created by adjusting each customer’s preannounced time window. This adjustment yields a presumed time window,  $(\bar{e}_i, \bar{l}_i)$ , where  $\bar{e}_i$  is obtained by adding a value  $\xi_i$  to  $\hat{e}_i$ , i.e.  $\bar{e}_i = \hat{e}_i + \xi_i$ . The value  $\xi_i$  is drawn from a distribution. For example, if the distribution of differences between the preannounced and real time windows was found to be a uniform distribution spanning a range of plus/minus 30 minutes, then  $\xi_i$  would be drawn from a uniform distribution with a range of 60 minutes. The closing time,  $\bar{l}_i$  is calculated by adding the length of the time window,  $TW_i$ , to this opening time. For each scenario, a route plan is generated by solving the off-line problem as described in section 3.1.

Each time  $c_i$  that a job confirms to the system, we update the set of scenarios  $S$ . The presumed time windows of the confirmed jobs are set to the confirmed time window in all scenarios. For the unconfirmed jobs whose presumed earliest pickup time,  $\bar{e}_i$ , has already passed, i.e.  $\bar{e}_i < t + L_i$ , we draw a new time window from the distribution while making sure that  $\bar{e}_i > t + L_i$ . Based on the updated time windows, we update the vehicles’ departure times for each plan and check whether the plan’s route sequence is still time feasible. If a plan is no longer feasible due to the updated time window scenario, we delete it and replace it by solving the off-line problem. Note that we can always obtain a feasible plan since the off-line problem can always reject a job if necessary.

Of the set of generated plans,  $\Pi$ , a distinguished plan  $\pi^*$  is selected by a consensus function. This plan is the plan that is parsed and enacted as the *operational* route plan. The distinguished plan represents the plan that is most similar to all others. The main idea is that the most similar plan encapsulates a least-commitment strategy (Stefik 1981) and as such is most likely to accommodate any given future realization. The consensus function scores the similarity of different route plans based on the immediate, next job after the job in execution at different points in time.

We first count the number of times that a job  $j$  is the immediate, next job in a route and the number of times that this immediate, next job  $j$  is assigned to a specific vehicle  $k$  as the immediate, next job, across all plans. We maintain these counts in a two dimensional matrix  $Y$  that is calculated for all plans each time we undertake a planning phase. Specifically, we compute a variable for all  $j \in J$ ,  $k \in V$ , and  $\pi \in \Pi$ :

$$y_{j,k,\pi} = \begin{cases} 1 & \text{if } j \text{ is the immediate, next job on vehicle } k \text{ in plan } \pi \\ 0 & \text{otherwise.} \end{cases}$$

We then use this variable to fill in the matrix  $Y$  with the sum of  $y_{j,k,\pi}$  over all  $\pi$  for each  $j$  and  $k$ . That is:  $Y_{j,k} = \sum_{\pi \in \Pi} y_{j,k,\pi}$ . Thus the matrix  $Y$  denotes the number of plans in which job  $j$  is served as the immediate, next job in the route of vehicle  $k$ . For each plan  $\pi$ , we can now define the vehicle consensus function as:

$$f_{seq}(\pi) = \sum_{j \in J} \sum_{k \in V} y_{j,k,\pi} \left( Y_{j,k} + \sum_{v \in V} Y_{j,v} \right). \quad (8)$$

The first term,  $y_{j,k,\pi} Y_{j,k}$ , within the consensus function summation is similar to Bent and Van Hentenryck (2004) and counts the number of plans that have the same job to vehicle pairs as in plan  $\pi$ . Our consensus function extends this count by including a sum over  $v$  of  $Y_{j,v}$  which counts the number of plans that have job  $j$  as the immediate, next job for any of the vehicle routes. This allows us to both consider the similarities in terms of the specific vehicle assignments and the route sequence, independent of the vehicle assignment.

Our scoring mechanism intrinsically handles job rejections since plans that consistently reject a job will be similar and thus have a higher score than those plans that include the job. Similarly, when the majority of plans include a job that the minority of plans reject, the distinguished plan will reflect the majority rather than the minority. This phenomenon is the basis of the example presented in the following paragraph. We select our distinguished plan  $\pi^* \in \Pi$  such that  $f_{seq}(\pi^*) \geq f_{seq}(\pi)$  for every  $\pi \in \Pi$ . We break ties arbitrarily by selecting the first of the highest scoring plans.

To illustrate the strengths of  $f_{seq}(\pi)$  relative to the vehicle based consensus function of Bent and Van Hentenryck (2004),  $f_{veh}(\pi) = \sum_{j \in J} \sum_{k \in V} y_{j,k,\pi} Y_{j,k}$ , we consider an example instance for which we generate five potential time realizations for all known, but not confirmed jobs at a decision epoch triggered by Job 2. The time realizations give rise to five plans over which each function operates. Table 1 provides the immediate, next assignment for each vehicle in each of the five plans. Additionally, Figure 4 shows the locations of the jobs and the current vehicle locations and highlights the interplay of the time scenarios with Job 1’s location relative to the vehicles. In the first time scenario, the presumed time window for Job 1 is set to a time before its preannounced time window and quite close to the current time. As a result, none of the vehicles can serve the job in time and the first plan is forced to reject Job 1. The remaining four plans all include Job 1 and place this job first on the route for one of the three available vehicles. However, dependent on the presumed time window scenarios for the other jobs, these four plans place Job 1 on different vehicles. We also note that in Time Scenario 5, as Vehicle 2 is the closest vehicle to Job 4, it is assigned to that more lucrative job at the expense of rejecting the confirmed Job 2. Furthermore, given the proximity of all three vehicles to Jobs 1, 2, and 5, Plans 2, 3, and 4, are essentially identical.

The last two columns of Table 1 identify the score associated with the vehicle based consensus function  $f_{veh}(\pi)$  and sequence based consensus function  $f_{seq}(\pi)$ , respectively. As  $f_{veh}(\pi)$  takes only the specific job to vehicle assignment into account when making the scores, the first plan – rejecting Job 1 – is selected. In contrast,  $f_{seq}(\pi)$  has more visibility into the plans’ collective ability to accommodate Job 1. As a result,  $f_{seq}(\pi)$  selects Plan 4. Note that enacting Plan 1 in this decision epoch does not imply that we actually reject Job 1 since we do not reject a job before its confirmation time. However, if, in the next decision epoch, Job 1 is confirmed then  $f_{veh}(\pi)$ ’s selected plan may force us to reject Job 1 as no vehicle was pre-positioned to serve it. The vehicles following the selected plan of  $f_{seq}(\pi)$  will, however, be positioned to act on Job 1’s confirmation. As job rejections are the most costly outcome,  $f_{seq}(\pi)$  gains the advantage over  $f_{veh}(\pi)$ .

Note, that if  $f_{seq}(\pi)$  did not include the first portion of Equation (8), that is the portion which is redundant relative to the job to vehicle assignments, then the scoring of Plans 1



Table 1: Immediate, next job for each vehicle and the rejected jobs across five plans; consensus function scores noted in last two columns – \* indicates selected plan.

	Vehicle 1	Vehicle 2	Vehicle 3	Reject	$f_{veh}(\pi)$	$f_{seq}(\pi)$
Plan 1	5	2	3	1	$2 + 2 + 2 = 6^*$	$6 + 6 + 4 = 16$
Plan 2	2	5	1	–	$1 + 1 + 1 = 3$	$5 + 5 + 5 = 15$
Plan 3	5	1	2	–	$2 + 1 + 1 = 4$	$6 + 5 + 5 = 16$
Plan 4	1	2	5	–	$2 + 2 + 1 = 5$	$6 + 6 + 5 = 17^*$
Plan 5	1	4	3	2	$2 + 1 + 2 = 5$	$6 + 2 + 4 = 12$

through 5 would be: 10, 12, 12, 12, and 7, respectively. As a result there would be a tie between Plans 2 - 4. However, given the similarity of the Plan 3 with the first plan, it should receive a higher similarity score. Thus, both portions of the  $f_{seq}(\pi)$  equation are important.

Each of the information strategies (*Ignore*, *Naïve*, and *MTS*) described here was tested on a set of test cases designed to mirror the structure and timings of the practical contexts where location information arrives before the time related information. The construction of these test cases is the topic of the next section.

## 5 Test Cases

To compare the different strategies, we generated a set of test cases based on transport data from a dial-a-chauffeur service in The Netherlands. These test cases may be accessed at <https://sites.google.com/site/pdptwinstances/>. The cost to the transport service provider is .30 euros/km for each kilometer driven; the revenue for each job includes a fixed fee of 6 euros plus 3 euros/km traveled in serving the job. Each instance consists of nine vehicles and a set of 20 jobs spread throughout a 100 square kilometer area with the depot located at the lower left corner, i.e. at [0,0]. We assume that vehicles travel in a straight line at unit speed. Each job has a given origin, destination, announcement time, confirmation time, and two time windows: preannounced and real.

The preannounced times,  $\hat{e}_i$ , are drawn from a uniform distribution spanning a six hour period of operations. The jobs' preannounced time windows,  $(\hat{e}_i, \hat{l}_i)$ , then follow from the preannounced times based on the use of fixed lead times and time window lengths; we assume

that both  $L_i$  and  $TW_i$  are the same for all  $i$ . To determine the confirmation times,  $c_i$ , we use a uniform distribution to draw deviations  $\xi_i$  from the interval  $[-30,30]$ . The jobs' real time windows then follow from adding a pre-announcement error  $\xi_i$  to the preannounced time window, i.e.  $(e_i, l_i) = (\hat{e}_i + \xi_i, \hat{l}_i + \xi_i)$ . This forms the base case with a range of 60 minutes with the preannounced time at the center of a uniform range of 60 minutes. This means that the average preannouncement error is zero (i.e. the preannounced time is an unbiased estimator of the real time) and that the pre-announced time estimate is at most 30 minutes earlier or later than the actual time.

As it is the time related information that brings both the uncertainty and interest to our problem, we develop a set of five test cases with varying levels of deviation in terms of the preannouncement errors. The errors are drawn from uniform ranges of 90, 120, 180, and 240 minutes. To allow for comparisons between the different cases, we fix the actual, confirmed times of the case with the smallest range, the Range60 case and vary the preannouncement errors around the actual time. As such, the Perfect Information benchmark is the same for all test cases. To increase the efficiency of our simulation, we reduce the variance between the results of the different experiments by making sure that all jobs that have a positive (negative) error in the Range60 case also have a positive (negative) error in the different test cases. However, the magnitude of the deviation is scaled for the longer ranges. Instead of drawing new preannouncement errors, we adjust the errors relative to the Range60 case. That is, the preannouncement error for a certain job  $i$  in the Range120 case is twice the error of that same job in the Range60 case just as the error for a job in the range 240 case would be four times that of the Range60 case, i.e.  $\xi_i^{120} = 2\xi_i^{60}$ .

We also establish a set of test cases around different lead time and time window lengths. Specifically, we set the lead times to 0, 5, 15, and 30 minutes and the time window lengths to 5, 15, and 30 minutes. The resulting time window cases were studied on a geography consistent with the dial-a-chauffeur service provider. In these cases, the preannounced time window is expected to deviate from the real time window within a uniform range of 120 minutes. The case with a lead time of five and a time window of five for a range of 120 in terms of the deviation was also used to test the efficacy of the scenario sampling strategies

when their underlying distributions are poorly calibrated. Table 2 provides a summary of the cases used in our experiments. Within each scenario we generated 100 instances: five different job location structures for each geography each with 20 different actual time window realizations.

The geography of our primary cases is one inspired by the dial-a-chauffeur service. However, to test the capabilities of our strategies we also studied two additional geographies. All three geographies are based on a 100 square kilometer region of operations. In this square, we have two cases that focus on the concept of a center region, much like a city center. The first case, is that which is consistent with a dial-a-chauffeur service provider serving a city – six jobs go from the center to outlying areas, four jobs go from outlying areas to the center, and ten jobs have random pickup and drop-off locations (BUS). The other case with a city center has all 20 jobs going from the center (in) to the outlying regions (out) (IO20). The third case has a fully random geography, with the jobs’ origins selected randomly and their corresponding destinations selected randomly (RR20). Figure 5 provides example instances for the BUS, IO20, and RR20 cases; customer origins are marked by a circle and the corresponding destination by a linked triangle. The higher the job id number, the later the job was confirmed.

## 6 Results

We implemented the simulation in C++ using Gurobi 5.6.3 as the MIP solver (Gurobi Optimization 2012) and ran all instances on an Intel i5 2.4 GHz computer. In our implementation of the MTS strategy, we maintain 60 plans (associated with 60 scenarios). The choice of 60 plans was driven by a trade-off between solution time and quality.

### 6.1 Comparison of Strategies

In the first set of experiments, we study the performance of the different strategies applied to all ranges of  $\xi_i$  with the announcement lead time  $L_i$  and time window length  $TW_i$  fixed at five

Table 2: Summary of instance parameters and default values.

Parameter	Definition	Possible values
$L_i$	Difference between the earliest service time $e_i$ and the confirmation time $c_i$ of job $i$	0, 5, 15, 30, or 60 min
$TW_i$	Difference between the earliest service time $e_i$ and the latest service time $l_i$ of job $i$	5, 10, 15, 30, 60 min
$\xi_i$	Difference between the earliest service time $e_i$ and the preannounced earliest service time $\hat{e}_i$ of job $i$	$U(-30, 30)$ , $U(-45, 45)$ , $U(-60, 60)$ , $U(-90, 90)$ , $U(-120, 120)$
Geography	The structure of the pickup and drop-off locations	BUS, IO20, RR20,

minutes each for all jobs. Table 3 presents the results for the different dynamic strategies and the Perfect Information benchmark. For the MTS strategy, we implement both the sequence-based consensus function (MTS-seq) and the vehicle-based consensus function (MTS-veh) as discussed in Section 4.2. We assess the performance of each strategy by computing the following statistics: the average, minimum, and maximum total costs relative to the Perfect Information case, the number of instances where this strategy performed the best in terms of the total costs (including ties), the average rejection costs, the average number of rejected jobs, the number of instances without rejections, and the average empty distance per served job. All of the averages are over 100 instances.

From Table 3, we observe that there is, on average, a 118% difference in costs between the Perfect Information benchmark and the Ignore strategy, which represents the value of perfect advance information. The results also indicate that the job information is beneficial, even if the time window information is only known probabilistically. We see that both the Naïve and MTS-seq strategies perform much better than the reactive Ignore strategy that disregards the preannounced information. The MTS-seq strategy consistently outperforms the other strategies both in terms of total costs and in terms of the number of instances for which it

gives the best solution. In particular, we see that it is the comparatively lower number of rejections and rejection costs that give the MTS-seq strategy an advantage over the other strategies including MTS-veh. The Naïve strategy and Ignore strategy have over twice as many rejections as the MTS-seq strategy. For the Ignore strategy, the higher rejections stems from the fact that the announcement lead time combined with the length of the time window (a total of 10 minutes in this setting) may not be enough for the vehicles to travel to the customer in time.

Focussing on the MTS strategies, if we compare the results for the different consensus functions, we see that the sequence-based consensus function, which puts additional weight on the service of a job on any vehicle performs best in terms of the average difference to the perfect information case as well as in the number and cost of rejections. This demonstrates the importance of using both the vehicle assignment and route position information for each job across all of the scenario based plans when the dimension of uncertainty is time.

Looking down the rows of Table 3, we note that the solution of the Perfect Information benchmark and the Ignore strategy do not depend on probabilistic information and are thus not susceptible to the increase in uncertainty. On the other hand, the results show that the performance of the Naïve and MTS strategies deteriorates with an increase in uncertainty. As the uncertainty increases, the difference between the reactive Ignore strategy and the proactive stochastic strategies decreases. When the uncertainty spans a uniform range of 240 minutes, the Ignore strategy outperforms the Naïve strategy and performs best in more instances than the MTS-veh strategy. This means that beyond a range of 240 minutes, it is better to ignore the information than to incorporate it in a naïve way. The reason for this is that the Naïve strategy moves towards a particular pickup location before the customer actually confirms his/her actual service time. While this may prevent job rejections if the customer confirms early, it may also make it impossible to serve another job. This is because with more temporal uncertainty, it is more likely that the sequence in which jobs call-in over time is different from the sequence of the preannounced times. Along with more rejections, this may result in more en route diversions as reflected in the empty distance. The MTS-seq strategy is more robust and deteriorates at a much slower rate than the Naïve strategy.

Even at high levels of uncertainty the MTS-seq strategy continues to perform significantly better than the other dynamic strategies including the myopic MTS-veh strategy. Furthermore, we note that in the most uncertain case the number of instances in which the Ignore strategy performed best is greater than those in which the MTS-veh strategy performed best. This is likely due to the generation of time scenarios that are far from accurate leading to the selection of plans that include job rejections as MTS-veh lacks the ability to assess the assignment of jobs to a variety of vehicles across the full set of plans.

## 6.2 Impact of Lead Time and Time Window Length

The results demonstrate that the preannounced time window information allows the service provider to serve customers within very short time windows with very short lead times. We now consider the impact of varying both the lead times and the time windows on the relative performance of the different strategies. In Figure 6(a), we present the total costs for all strategies across five different lengths of lead time (0, 5, 15, 30, 60), with the time windows set to five minutes. In Figure 6(b), we present the total costs of all strategies applied to four different lengths of time windows (5, 15, 30, 60) at the mid-range

As expected, we see that both the longer lead times and the longer time windows have a positive effect on the performance of the different strategies. Although the impact is similar, there is a subtle difference in the way they help the strategies cope in the face of uncertainty. A longer lead time implies an earlier resolution of uncertainty, while longer time windows provide more flexibility to serve the job once the uncertainty has been resolved. The results show that an increase in the time window length provides greater performance gains than the same increase in lead time length. This is intuitive since longer time windows allow for more flexibility in the assignment of jobs to vehicles. These results are consistent with the literature where we see that having a release time as opposed to a time window allows for more freedom in the sequence that jobs are visited and thus alleviates costs associated with the time dimension (Desrosiers et al. 1995). It is interesting to note that the performance gap between the different dynamic strategies decreases with both the longer lead times and the

Table 3: Results for routing strategies applied to Range120, with 5 minutes of lead time and 5 minute time windows, averaged across 100 instances.

	% $\Delta$ Perfect Info.			# Best	Avg. Rejection Costs	# of Rejections	# Inst. with No Rejections	Empty Dist. per Job Served
	Avg. Cost	Min. Cost	Max. Cost					
<b>Range60;</b> $\xi_i \in \mathbf{U}(-30, 30)$								
Perfect	0.0	0.0	0.0	–	13.0	0.2	82	68.8
Ignore	118.0	23.4	257.3	2	552.4	4.5	0	78.3
Naïve	39.1	0.0	166.0	33	170.4	1.6	17	75.2
MTS-veh	26.4	0.0	108.5	30	89.1	.9	40	77.2
MTS-seq	24.0	1.2	102.3	46	77.2	.7	42	77.0
<b>Range90;</b> $\xi_i \in \mathbf{U}(-45, 45)$								
Perfect	0.0	0.0	0.0	–	13.0	0.2	82	68.8
Ignore	118.0	23.4	257.3	1	552.4	4.5	0	78.3
Naïve	51.8	0.0	166.0	20	226.0	2.0	9	76.6
MTS-veh	39.5	0.0	140.5	33	145.9	1.3	20	78.9
MTS-seq	32.9	0.0	99.5	50	109.2	1.0	36	79.0
<b>Range120;</b> $\xi_i \in \mathbf{U}(-60, 60)$								
Perfect	0.0	0.0	0.0	–	13.0	0.2	82	68.8
Ignore	118.0	23.4	257.3	4	552.4	4.5	0	78.3
Naïve	65.2	0	164.0	14	282.3	2.5	6	78.8
MTS-veh	50.6	2.0	128.3	42	190.4	1.6	17	80.7
MTS-seq	44.0	2.3	136.9	50	158.5	1.4	25	80.4
<b>Range180;</b> $\xi_i \in \mathbf{U}(-90, 90)$								
Perfect	0.0	0.0	0.0	–	13.0	0.2	82	68.8
Ignore	118.0	23.4	257.3	9	552.4	4.5	0	78.3
Naïve	95.5	6.1	235.1	17	421.6	3.7	3	82.4
MTS-veh	67.5	3.5	173.8	33	267.6	2.2	7	82.1
MTS-seq	60.5	7.8	183.8	44	226.8	1.9	9	82.7
<b>Range240;</b> $\xi_i \in \mathbf{U}(-120, 120)$								
Perfect	0.0	0.0	0.0	–	13.0	0.2	82	68.8
Ignore	118.0	23.4	257.3	28	552.4	4.5	0	78.3
Naïve	127.4	5.3	283.9	10	571.1	4.9	1	85.7
MTS-veh	96.7	10.7	282.7	22	400.5	3.3	1	85.1
MTS-seq	88.0	10.1	221.0	41	349.5	2.8	1	85.9

longer time windows. This is to be expected as a very long time window obfuscates the need for a strategy that explicitly incorporates the uncertainty of the time window information since there is more time to accommodate jobs once confirmed.

When we consider the impact of uncertainty across the various time window lengths, the performance of the dynamic strategies deteriorates with an increase in uncertainty. This can be seen for both the Naïve and MTS-seq strategy in Figure 7. The competitive advantage of MTS-seq over the Naïve strategy is stable over the various uncertainty levels. We also see that the length of the time windows has a damping effect on the impact of uncertainty. Specifically, the cost increase for the MTS-seq strategy between the lowest level of uncertainty (Range60) and the highest level of uncertainty (Range240) is 52% for the five minute time windows, while it is only 29% for the 30 minute time windows.

### 6.3 Impact of Geography

Finally, in order to demonstrate the MTS-seq strategy’s capabilities across a wide range of operating conditions, we study the results of the dynamic strategies and the Perfect Information Benchmark as applied on three different geographies. In each of these geographies, the level of uncertainty spans a range of 120 minutes and both the lead time and time windows are five minutes. Note that while the job pickup and drop-off locations vary widely across the different geographies, the real time windows set for the jobs are the same from case to case. That is, all of the jobs in the first instance of the BUS case have the same time windows as the first instance of the RR20 case, and so on. This allows us to isolate the impact of time relative to geographic structure. The results of these simulations are depicted in Figure 8 which shows the minimum, first quartile, median, third quartile, and maximum value of the routing costs (a) and the rejection costs (b) across the 100 instances for each geography.

From Figure 8(a), we see that all strategies perform similarly in terms of the routing costs across the three geographies. The stochastic MTS strategies exhibit slightly higher routing costs than the Naïve strategy in all geographies and a much higher range of routing costs in the I020 case. In contrast, the MTS strategies show consistently lower rejection costs



than the Naïve strategy across all geographies. Furthermore, the MTS-seq strategy shows a smaller range of rejection costs compared to the MTS-veh strategy in both the BUS and IO20 cases. In the RR20 case, the MTS-seq and MTS-veh strategies perform similarly.

These results are instructive as they highlight the spatial structure that gives MTS-seq its competitive advantage in the face of temporal uncertainty. Specifically, Figure 8(b) shows that in both the BUS and IO20 cases, MTS-seq has lower rejection costs than MTS-veh. One potential reason for this is that explicitly taking into account the route sequence across all plans is beneficial in situations in which more than one immediate, next job to vehicle assignment is possible across the range of time scenarios. These situations arise when multiple vehicles are in a region with multiple jobs such as the city center in the BUS and IO20 geographies. This was illustrated in the example provided in Section 4.2. In RR20, however, the random location of jobs yields few if any regions with multiple jobs and thus MTS-seq loses its competitive advantage.

## 6.4 Impact of Wrong Input Parameters

In the experiments presented thus far, the MTS strategies used the same probability distribution for the preannouncement errors as those used when creating the problem instances. However, in a real-world setting the true distributions are unknown and need to be approximated based on historical data. Figure 9 shows how the MTS strategies perform for the Range120, BUS instances, using the wrong range parameter for the uniform distribution of the errors, with values ranging between 60 (representing an underestimate) and 240 (representing an over estimate). As a benchmark, we also plot the Naïve strategy. As before, we see that both MTS strategies outperform the Naïve strategy in all cases and the MTS-seq consistently outperforms the MTS-veh strategy. Figure 9 highlights the MTS-seq strategy’s ability to perform consistently across a wider range of incorrect calibration (60 to 180) than MTS-veh. This is an important result as it suggests that in realistic settings, in which we have to approximate the probability distributions, the MTS strategy outperforms strategies that do not explicitly incorporate the uncertainty.

## 7 Conclusion

In this paper, we studied a dynamic pickup and delivery problem with time windows in which the job locations are known in advance, but precise information on the desired service time windows is lacking. Only during operations, close to the desired pickup-time, does the actual time window information become available. We proposed several strategies to cope with uncertain advance time window information in this setting and compared their relative merits in terms of routing costs.

Based on our experiments, we show that advance information, even if uncertain, can provide benefits from a planning perspective. In settings with only moderate levels of uncertainty, the Naïve strategy, which treats the uncertain information as certain, outperforms the Ignore strategy, which ignores information until all uncertainty is resolved. Only when the uncertainty is large (greater than  $\pm 120$  minutes) does it become better to employ the purely reactive Ignore strategy. The stochastic mechanism that we introduced, the MTS strategy with the sequence based consensus function, is consistently robust. It outperforms both the Naïve and the Ignore strategies, in all cases, even if we do not know the true probability distribution that characterizes the uncertainty.

Of most interest managerially, is the recognition that having customers preannounce their need for service, even when the time related information is not completely accurate, allows the service provider to offer shorter time windows and shorter lead times. This is especially true when using a real-time strategy that explicitly incorporates the uncertainty of this information, such as the MTS-seq strategy, to support the route planning. The MTS-seq strategy exhibits fewer numbers and costs of rejections across a variety of cases with very short time windows and lead times.

The strength of MTS-seq lies in its extension of the original consensus function of Bent and Van Hentenryck (2004) by not only scoring the similarity in terms of the job to vehicle assignments but also in terms of the route sequence. We show that this additional component is beneficial in handling the uncertainty that characterizes our specific problem setting. Nevertheless, additional experiments are required to investigate to what extent this

extension is beneficial to vehicle routing problems in general and to those that Bent and Van Hentenryck (2004) were solving, in specific. Future work could also consider the value of (adaptively) varying the weights on these two census function components across different problem settings.

As this is one of the first papers that isolates the impact of temporal uncertainty in the context of pickup and delivery problems, we see many opportunities for further research. One opportunity for future study lies in the use of ADP to solve this problem. While the current modeling from the perspective of vehicles serving jobs makes the study of the state space complex, an alternative view of jobs vying for vehicles may yield a valuable strategy.

We also recommend considering more sophisticated waiting strategies to determine when and where to wait given the probabilistic information on the required timing of subsequent jobs. That is, instead of moving to a particular job location or waiting at the current location, it may be beneficial to move the vehicles to “promising” locations from which several potential jobs could be served in time.

While we have exclusively focussed on a unit-capacity, one-to-one pickup and delivery setting, pre-announcing the need for service without committing to a specific time window is also common in less-than-truckload package pickup and delivery problems. These settings can typically be characterized as one-to-many or many-to-one pickup and delivery problems. While the general idea of our strategies would theoretically also apply here, it would be more difficult to find optimal solutions to the off-line subproblem within reasonable times.

When it becomes more time-consuming to solve the subproblems, it may be relevant to investigate whether we can reduce the number of plans in our sample-based strategy and then alter the corresponding scoring mechanism to provide routes efficiently without deteriorating the solution quality. Instead of selecting a particular, most similar plan from the set of plans, it could, for example, be beneficial to create a distinguished plan by combining the best scoring parts of different plans. Creating such a plan, while maintaining feasibility and assuring a certain level of consistency across the different planning epochs, is a challenging area for future research.

## References

- Bandyk, Matthew. 2012. Designated drivers a nascent industry. URL <http://americasfuture.org/doublethink/2012/11/designated-drivers-a-nascent-industry/>.
- Bent, R.W., P Van Hentenryck. 2004. Scenario-Based Planning for Partially Dynamic Vehicle Routing with Stochastic Customers. *Operations Research* **52**(6) 977–987.
- Berbeglia, Gerardo, Jean-François Cordeau, Gilbert Laporte. 2010. Dynamic pickup and delivery problems. *European Journal of Operational Research* **202**(1) 8–15.
- Cortes, C. E., D. Saez, a. Nunez, D. Munoz-Carpintero. 2009. Hybrid Adaptive Predictive Control for a Dynamic Pickup and Delivery Problem. *Transportation Science* **43**(1) 27–42. URL <http://transci.journal.informs.org/cgi/doi/10.1287/trsc.1080.0251>.
- Desrosiers, Jacques, Yvan Dumas, Marius M Solomon, François Soumis. 1995. Time constrained routing and scheduling. *Handbooks in operations research and management science* **8** 35–139.
- Dror, Moshe, Gilbert Laporte, Pierre Trudeau. 1989. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation science* **23**(3) 166–176.
- Ghiani, Gianpaolo, Emanuele Manni, Barrett W Thomas. 2012. A comparison of anticipatory algorithms for the dynamic and stochastic traveling salesman problem. *Transportation Science* **46**(3) 374–387.
- Goodson, Justin C, Jeffrey W Ohlmann, Barrett W Thomas. 2013. Rollout policies for dynamic solutions to the multivehicle routing problem with stochastic demand and duration limits. *Operations Research* **61**(1) 138–154.
- Gurobi Optimization, Inc. 2012. Gurobi optimizer reference manual. URL <http://www.gurobi.com>.
- Hvattum, Lars M., Arne Løkketangen, Gilbert Laporte. 2006. Solving a Dynamic and Stochastic Vehicle Routing Problem with a Sample Scenario Hedging Heuristic. *Transportation Science* **40**(4) 421–438.
- Hyttiä, Esa, Aleksi Penttinen, Reijo Sulonen. 2012. Non-myopic vehicle and route selection in dynamic darp with travel time and workload objectives. *Computers & Operations Research* **39**(12) 3021–3030.

- Ichoua, Soumia, Michel Gendreau, Jean-Yves Potvin. 2006. Exploiting Knowledge About Future Demands for Real-Time Vehicle Dispatching. *Transportation Science* **40**(2) 211–225.
- Jaillet, Patrick, Xin Lu. 2011. Online traveling salesman problems with service flexibility. *Networks* **58**(2) 137–146. doi:10.1002/net.20454. URL <http://dx.doi.org/10.1002/net.20454>.
- Jaillet, Patrick, Michael R. Wagner. 2006. Online routing problems: Value of advanced information as improved competitive ratios. *Transportation Science* **40**(2) 200–210. doi:10.1287/trsc.1060.0147.
- Kim, Yongjin, Hani S Mahmassani, Patrick Jaillet. 2004. Dynamic truckload routing, scheduling, and load acceptance for large fleet operation with priority demands. *Transportation Research Record: Journal of the Transportation Research Board* **1882**(1) 120–128.
- Larsen, Allan, Oli BG Madsen, Marius M Solomon. 2004. The a priori dynamic traveling salesman problem with time windows. *Transportation Science* **38**(4) 459–472.
- Mes, Martijn, Matthieu van der Heijden, Peter Schuur. 2010. Look-ahead strategies for dynamic pickup and delivery problems. *OR spectrum* **32**(2) 395–421.
- Mitrović-Minić, Snežana, Ramesh Krishnamurti, Gilbert Laporte. 2004. Double-horizon based heuristics for the dynamic pickup and delivery problem with time windows. *Transportation Research Part B: Methodological* **38**(8) 669–685.
- Parragh, Sophie N., Karl F. Doerner, Richard F. Hartl. 2008. A survey on pickup and delivery problems. *Journal für Betriebswirtschaft* **58**(2) 81–117.
- Pillac, Victor, Michel Gendreau, Christelle Guéret, Andrés L Medaglia. 2012a. A review of dynamic vehicle routing problems. *European Journal of Operational Research* **225**(1) 1–11.
- Pillac, Victor, Christelle Guéret, Andrés Medaglia. 2012b. An event-driven optimization framework for dynamic vehicle routing. *Decision Support Systems* **54**(1) 414–423.
- Powell, Warren B., Yosef Sheffi, Kenneth S. Nickerson, Kevin Butterbaugh, Susan Atherton. 1988. Maximizing profits for north american van lines' truckload division: A new framework for pricing and operations. *Interfaces* **18**(1) pp. 21–41. URL <http://www.jstor.org/stable/25061046>.
- Savelsbergh, M.W.P., M. Sol. 1995. The general pickup and delivery problem. *Transportation science* **29**(1) 17–29.
- Srour, Jordan, Rob Zuidwijk. 2008. How much is location information worth? a competitive

- analysis of the online traveling salesman problem with two disclosure dates. Tech. Rep. ERS-2008-075-LIS, E. URL <http://hdl.handle.net/1765/13837>.
- Stefik, Mark. 1981. Planning with constraints (molgen: Part 1). *Artificial intelligence* **16**(2) 111–139.
- Thomas, Barrett W. 2007. Waiting strategies for anticipating service requests from known customer locations. *Transportation Science* **41**(3) 319–331.
- Thomas, Barrett W., Chelsea C. White. 2004. Anticipatory Route Selection. *Transportation Science* **38**(4) 473–487.
- Williams, WH, ML Goodman. 1971. A simple method for the construction of empirical confidence limits for economic forecasts. *Journal of the American Statistical Association* **66**(336) 752–754.
- Yang, J., P. Jaillet, H. Mahmassani. 1999. On-line algorithms for truck fleet assignment and scheduling under real-time information. *Transportation Research Record* **1667** 107–113.
- Yang, J., P. Jaillet, H. Mahmassani. 2004. Real-time multivehicle truckload pickup and delivery problems. *Transportation Science* **38**(2) 135–148.

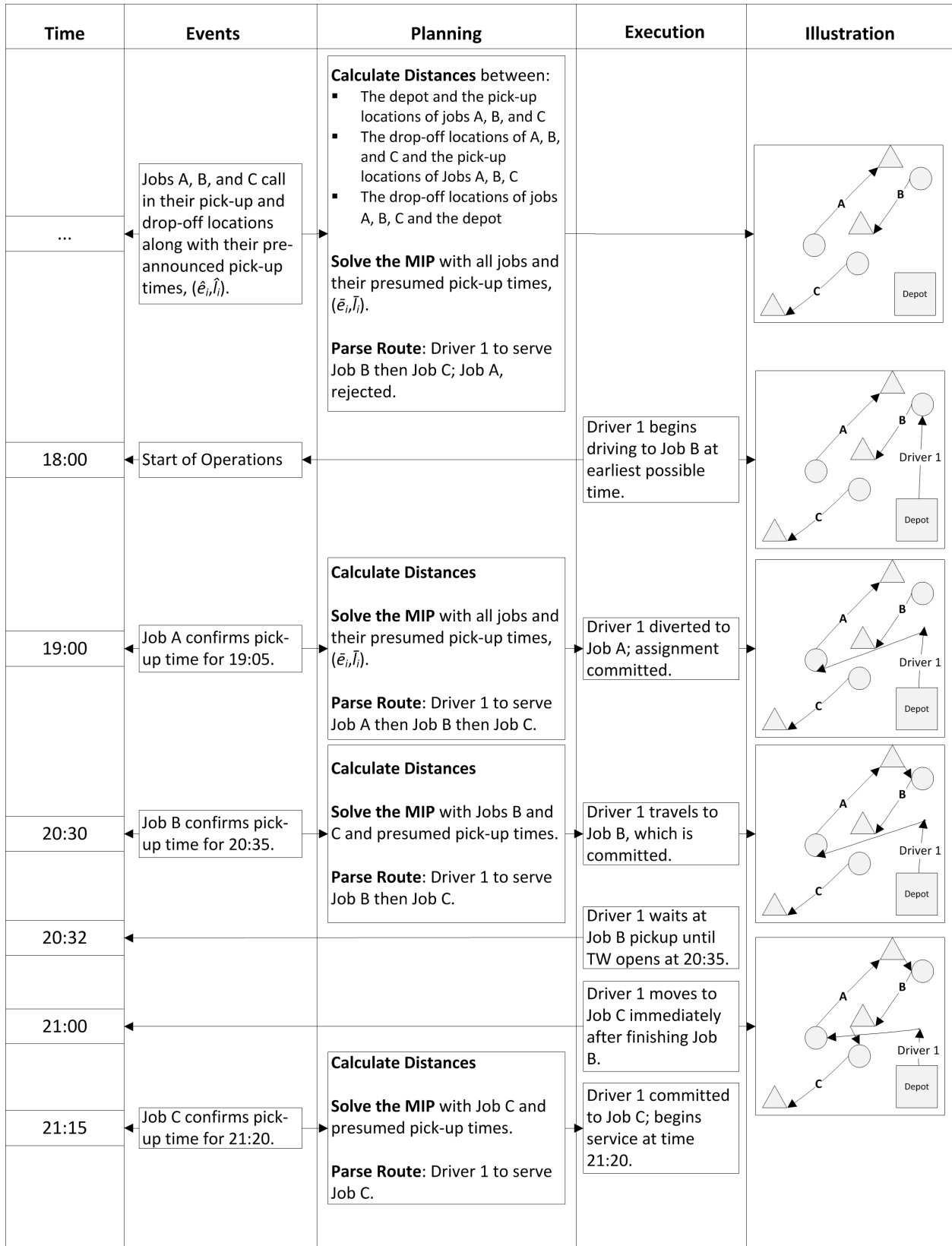


Figure 3: The method by which the off-line problem (the MIP) is solved as part of a rolling horizon framework.

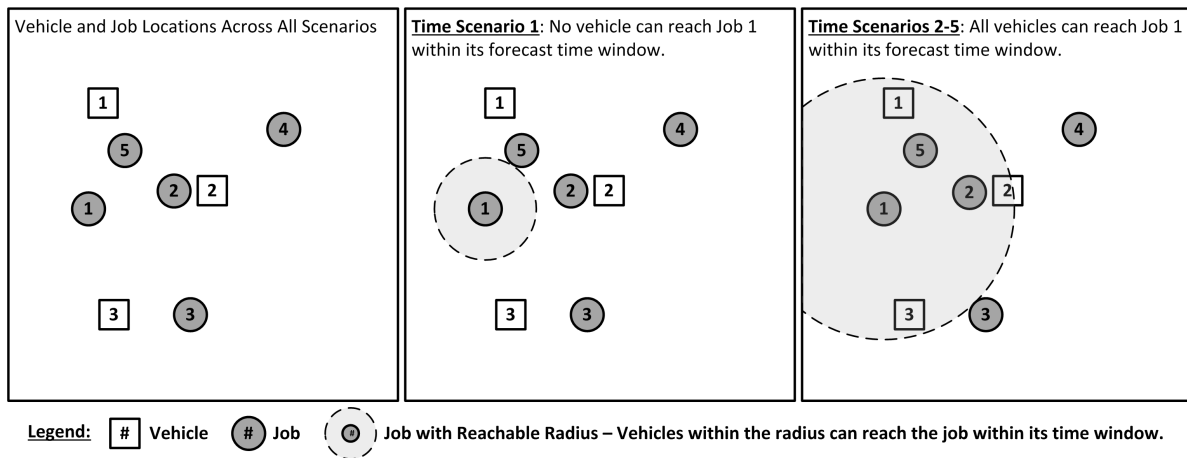
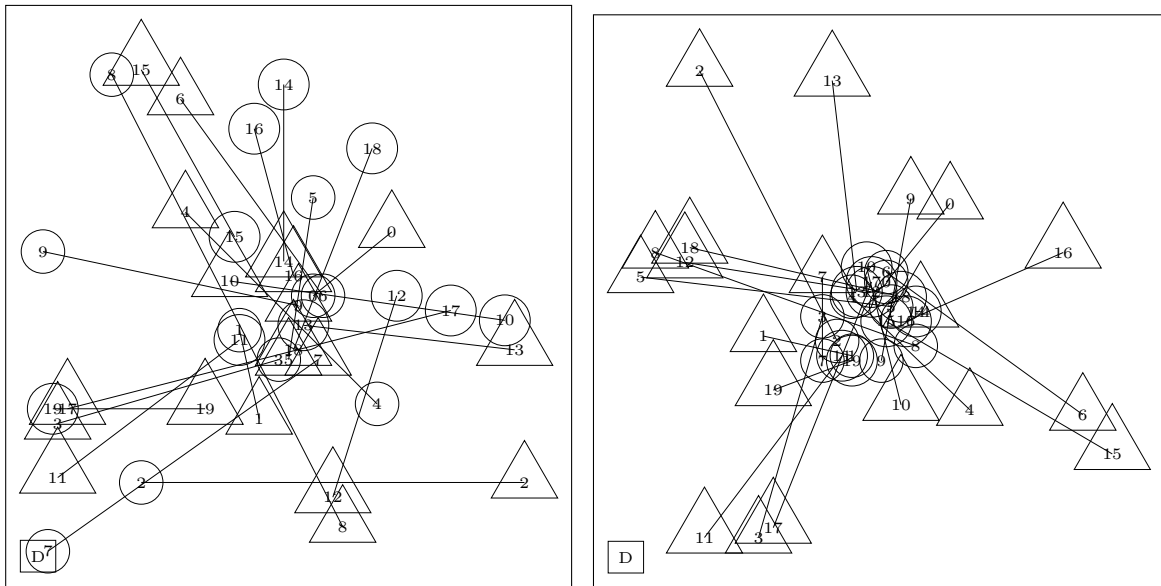


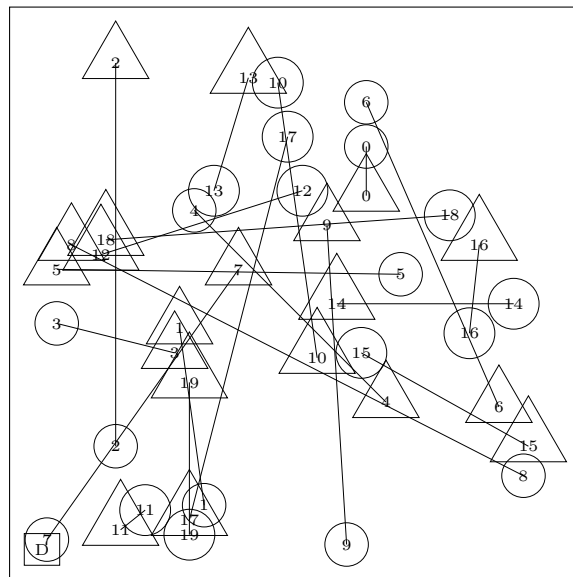
Figure 4: Depiction of time scenarios and the interplay with geography to yield route plans.





(a) BUS

(b) IO20



(c) RR20

Figure 5: Example instances with customer origins (circle) and destinations (triangle) and the depot (rectangle).

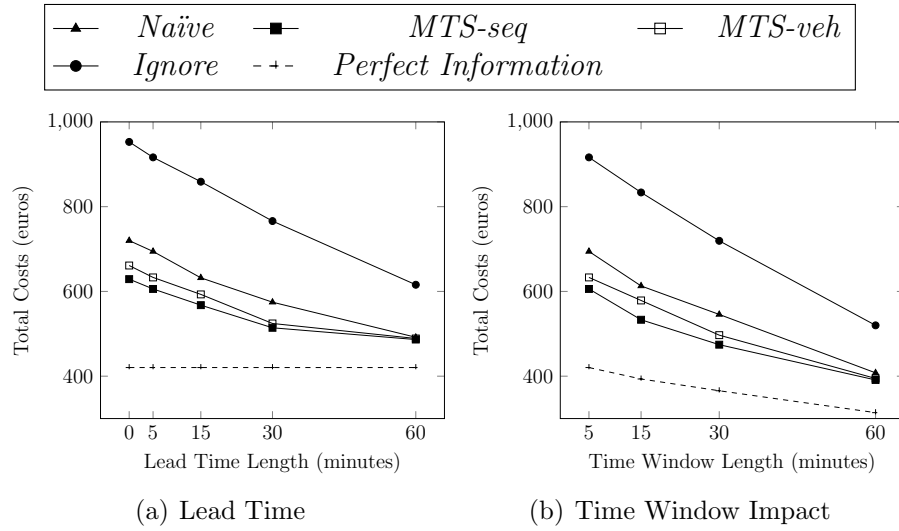


Figure 6: Impact of different lead times and time window lengths on the total costs of routing strategies applied to the Range120 case;  $n = 100$ .

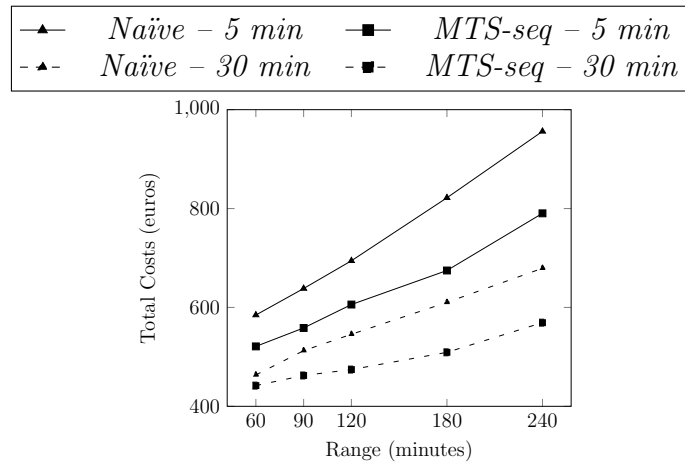
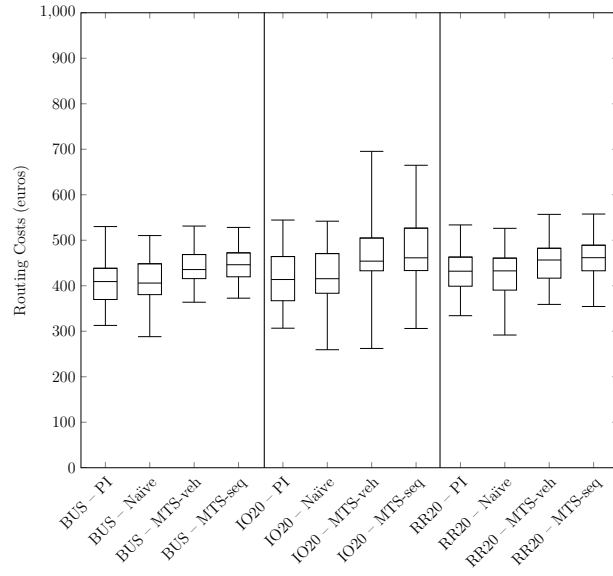
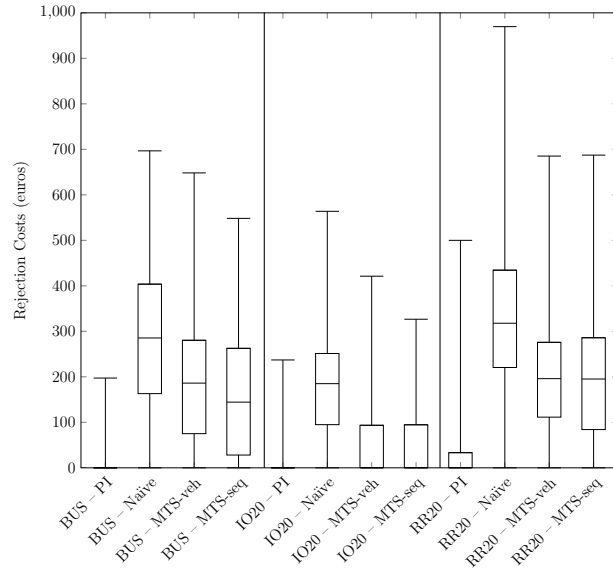


Figure 7: Total costs for the Naïve and MTS-seq strategies across different ranges of uncertainty with time windows lengths of 5 minutes (solid) and 30 minutes (dashed);  $n=100$



(a) Routing Costs



(b) Rejection Costs

Figure 8: Minimum, 1st quartile, median, 2nd quartile, and maximum routing costs (a) and rejection costs (b) for three routing strategies and the perfect information benchmark across three geographies;  $n = 100$ .

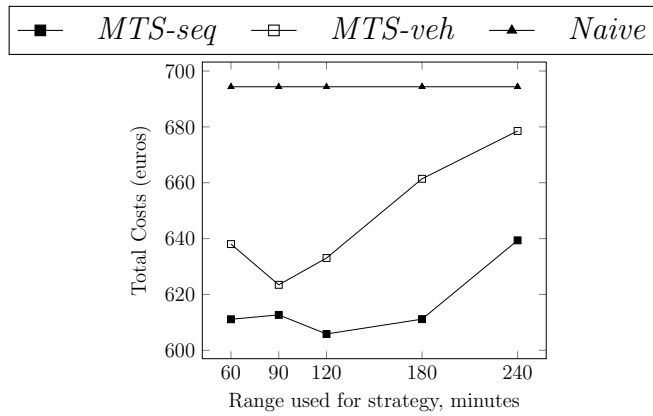


Figure 9: Total costs for Range120 instance with different ranges used in sample scenarios, n=100

<b>ERIM Report Series <i>Research in Management</i></b>	
ERIM Report Series reference number	ERS-2014-015-LIS
Date of publication	2014-11-24
Version	12-08-2015
Number of pages	37
Persistent URL for paper	<a href="http://hdl.handle.net/1765/77187">http://hdl.handle.net/1765/77187</a>
Email address corresponding author	nagatz@rsm.nl
Address	Erasmus Research Institute of Management (ERIM) RSM Erasmus University / Erasmus School of Economics Erasmus University Rotterdam PO Box 1738 3000 DR Rotterdam, The Netherlands Phone: +31104081182 Fax: +31104089640 Email: <a href="mailto:info@erim.eur.nl">info@erim.eur.nl</a> Internet: <a href="http://www.erim.eur.nl">http://www.erim.eur.nl</a>
Availability	The ERIM Report Series is distributed through the following platforms: RePub, the EUR institutional repository Social Science Research Network (SSRN) Research Papers in Economics (RePEc)
Classifications	The electronic versions of the papers in the ERIM Report Series contain bibliographic metadata from the following classification systems: Library of Congress Classification (LCC) Journal of Economic Literature (JEL) ACM Computing Classification System Inspec Classification Scheme (ICS)