The Value of Inaccurate Advance Time Window Information in a Pick-up and Delivery Problem

F. Jordan Srour[*]
Lebanese American University,
P.O. Box 13-5053, Chouran, Beirut 1102 2801, Lebanon
{jordan.srour@lau.edu.lb},
Niels Agatz
Rotterdam School of Management, Erasmus University,
Burg. Oudlaan 50, 3062 PA Rotterdam, The Netherlands
{nagatz@rsm.nl},
Johan Oppen
Molde University College,
P.O.box 2110, N-6402 Molde, Norway
{Johan.Oppen@hiMolde.no}

Abstract

We examine different routing strategies to cope with inaccurate time window information in the context of a dynamic pick-up and delivery problem with time windows. Our experiments show that advance information, even if inaccurate, can provide benefits from a planning perspective. We propose a novel stochastic strategy that consistently performs well compared to several benchmark strategies.

Keywords: Routing; Value of Information; Dynamic Pick-up and Delivery Problem with Time Windows; Multiple Scenario Analysis

1 Introduction

There is wide recognition that most vehicle routing problems encountered in practice are dynamic. In a dynamic setting, a dispatcher must make routing decisions in one moment

[*] Both Drs. Srour and Agatz contributed equally to this publication. This research was partially supported by a grant from the Lebanese American University. The authors also thank Dr. Leo Kroon for his helpful comments on a draft of this work.

that may become obsolete in the next. To hedge against obsolescence, we study strategies for handling inaccurate information in the context of a dynamic pick-up and delivery problem with time windows. In particular, we study a setting where the job locations are known, but their exact service time windows are only revealed over time. We do this in the context of a designated driver service that drives people home in their own vehicles after a night out.

Although the numbers are steadily declining, alcohol-impaired driving remains a major safety issue worldwide. In the United States, 9,878 people were killed in alcohol-impaired-driving crashes in 2011 (NHTSA, 2012). These alcohol-impaired driving fatalities account, annually for approximately one third of the total motor vehicle traffic fatalities in the United States; similar numbers hold for Europe (ETSC, 2008). To provide a safe and convenient alternative to driving home inebriated, a variety of companies offer designated driver services. With this concept, the customer requests a chauffeur to transfer him/her (and possibly other passengers) from a bar or restaurant to his/her home in his/her own car.

While the concept originates in Asia (Sang-Hun, 2007), several companies in Europe and the US now offer such a service. Recent startup Stearclear, in the United States, offers a smart phone application supported by GPS technology to facilitate and automate the booking process for the service. Rentabob, Superbob and Beterbob are examples of companies that offer designated driver services in the Netherlands. (Bob is a colloquial term in the Benelux that refers to the designated driver.)

The general method of operation for these designated driver companies is through the use of chauffeurs and company cars. In fact, the franchising information of Stearclear recommends the use of teams – one driver for a company car and one chauffeur for the customers. As depicted in Figure 1, the chauffeur arrives to the customer pick-up location by a company car. The chauffeur then uses the customer's car to take the customer to his/her destination. After arriving with the customer at the destination, the chauffeur is then picked up by a company car to travel to the next customer pick-up location. Customers pay an initial fee plus a distance related fee. The fees are typically higher than a regular taxi, but cheaper than taking a round-trip taxi ride, which would be required if the customer takes a taxi back home and goes back the next day to pick-up his/her car. The problem is dynamic as the job requests are not fully known in advance.

Companies usually allow customers to place pre-announcements that enable them to reserve a chauffeur in advance without initially committing to a specific time. When pre-announcing the customer will quote both the pick-up and drop-off location along with

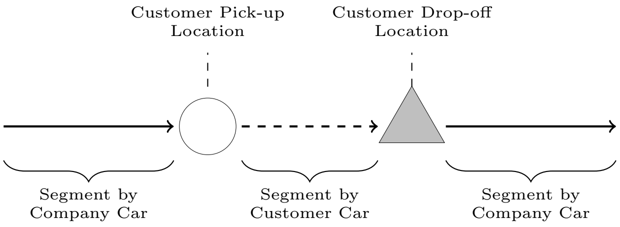


Figure 1: Example routing for one chauffeur and one customer.

an approximate pick-up time, e.g. “around 11pm”. Later, during operations, when the pre-announced customer actually requires service, they call the company for pick-up. If announced early enough, pre-announcements are beneficial to support decisions on how many company cars to deploy on a given day. However, it is unclear if inaccurately pre-announced service time information is beneficial during operations. This leads to the fundamental question addressed in this paper: when does inaccurate advance time window information provide value in the context of dynamic routing operations?

1.1 Related Literature

The value of a decision made in an uncertain environment is largely influenced by two factors: the quality of the information at the time of decision making and the future realized state of the environment. The relationship between information quality and “good” decision making is well recognized in many fields. The field of Transportation is no different; with the growing interest in dynamic or real-time vehicle routing strategies, there is a growing interest in the value of information (Psaraftis, 1995; Gendreau and Potvin, 1998; Ghiani et al., 2003; Mitrović-Minić et al., 2004; Thomas, 2010; Pillac et al., 2012). What separates Transportation from other fields is the simultaneous spatial and temporal nature of information in vehicle routing. In dynamic routing problems, jobs that require service by a vehicle or a fleet of vehicles arise during operations at specific locations and often require service at specific times. As noted by Psaraftis (1995), the information about both time and location evolves over time – with accuracy at its worst during initial planning and at its best during operations. This is the case for the designated driver problem as customer requests continuously arrive and are updated over time; with only the final confirmation, during operations, being fully reliable.

Coping with this evolution of information can be done in three ways: through the incorporation of stochastic information (Dror et al., 1989), through update methods, where a static problem is solved repeatedly (Berbeglia et al., 2010; Pillac et al., 2012), or through stochastic, dynamic strategies that combine both approaches. Due to the fundamental importance of job locations in routing problems, most dynamic stochastic strategies focus on forecasting future job locations. Thomas and White (2004) consider a single vehicle that travels between a given origin and a given destination where new service requests may arrive from potential customers while the vehicle is in transit. They model the problem of determining a route that anticipates these potential future pick-up requests along the way. Schilde et al. (2011) consider a dial-a-ride system that involves the transportation of patients to and from hospitals. For each trip to the hospital, there is a corresponding return trip home on the same day with a certain probability. They show that taking into account stochastic information about future return transports is beneficial for the resulting solution quality. They present different metaheuristic approaches to solve this problem. Hyytiä et al. (2012) consider a policy that explicitly takes into account the unknown future requests by using a Markov decision framework. Cortes et al. (2009) present a hybrid predictive control framework to model the impact of stochastic rerouting delays for dynamic multi-vehicle pick-up and delivery problems.

Of most relevance to our work is the Multiple Scenario Analysis (MSA) work of Bent and Van Hentenryck (2004). In the MSA strategy, at each decision epoch, a set of potential plans is generated to match a set of potential job realizations. From among these plans, the plan with the most similarity to all of the other plans, in terms of vehicle to job assignments, is selected and enacted. Ghiani et al. (2012) provide a recent application of this approach in the context of a dynamic and stochastic traveling salesman problem. In our work, we also introduce a strategy that selects a plan from a set of potential plans. However, the set of plans we generate are not for potential spatial job realizations, but for potential temporal job realizations. We further adapt the MSA strategy by introducing a plan selection function that explicitly accounts for both the next job and the vehicle to which that next job is assigned.

While this paper focuses on the use of advance information, we cannot ignore the underlying routing structure. We assume that our routing problem follows the current operational practices of most designated driver companies, where each company car is paired to one chauffeur and the company car follows the chauffeur as s/he drives the customer home. Thus, the matching of chauffeurs to customers resembles a pick-up and delivery problem with time windows associated with the pick-up locations. Focusing on the pick-up and delivery literature, as noted in Parragh et al. (2008), a host of articles measure the impact of real-time routing in the single vehicle, dynamic pick-up 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 and multiple chauffeurs. Of the literature focused on the multi-vehicle dynamic pick-up and delivery problem, Mitrović-Minić et al. (2004) is a valuable reference as they introduce a metric termed the value of information (a ratio of the difference between an online and off-line objective value to the online objective value). They use this metric to evaluate 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. In contrast to the PDPTW problem of Mitrović-Minić et al. (2004), we are interested in PDPTWs where each job requires the full capacity of the vehicle, since each chauffeur can handle only one customer at a time. As these problems most frequently arise in freight transportation, the problem class is termed the dynamic truckload pick-up and delivery problem (Savelsbergh and Sol, 1995). In the dynamic truckload, pick-up and delivery problem literature, the work of Yang et al. is most relevant to our work.

In Yang et al. (1999), the authors introduce an efficient mixed-integer program for use in a rolling horizon strategy. They demonstrate that using the 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 pick-up and drop-off locations outperforms the simple reactive strategy. In this paper, we adapt the use of Yang et al.'s mixed integer program to handle pre-announced jobs with potentially unreliable time window information.

The presence of pre-announced jobs in the designated driver problem also puts us in a good position to leverage strategies for incorporating stochastic information in route planning. 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). In the realm of truckload transport, Tjokroamidjojo et al. (2006) quantify the value of advance job information showing that significant cost savings are possible when a company receives accurate information about a job in advance, but can delay placing the job on a specific route until the latest possible moment. While this quantification of advance information is interesting, we seek to extend this work by examining the value of advance job-related information when its accuracy is questionable.

1.2 Our Contributions

In addition to presenting the designated driver problem, this paper adds to the literature on the value of advance information by studying the effect of inaccurate advance time window information on operations. To the best of our knowledge, this is the first paper to study the impact of inaccurate advance time window information in vehicle routing.

Building on the work of Bent and Van Hentenryck (2004), we propose the *Multiple Time Scenario* (*MTS*) strategy, a novel online routing strategy to deal with inaccurate time window information. This strategy generates a set of scenarios with different timing realizations and then selects a candidate route relative to all the routes generated for these time-based scenarios. We demonstrate the efficacy of the *MTS* strategy through experiments based on data from a real-world application. Our results highlight the capability of the *MTS* strategy in extracting value from even the most inaccurate advance information.

The next section describes the assumptions underlying our version of the designated driver problem along with our formulation of the static routing problem, its use in a rolling horizon framework, and the routing strategies that we test – with a particular emphasis on the *MTS* strategy. In Section 3, we describe the construction and structure of the problem instances used to test the strategies. In Section 4, we present the results of the computational experiments. We conclude the paper in Section 5 with a discussion of our findings from both a theoretical and managerial perspective.

2 The Designated Driver Problem

The designated driver problem requires the assignment of chauffeurs to customers and company cars to chauffeurs. At the moment that customer requests arrive, the available fleet of company cars is fixed. We use the term company car to refer to an employee and car combination whose job is to drive chauffeurs. We use the term chauffeur to refer exclusively to the person who drives the customer and the customer's car. If we consider fixed chauffeur-company car pairs, the problem reduces to assigning the chauffeur-company car pairs to customer requests. This, in turn, represents a dynamic truckload pick-up and delivery problem with time windows.

Let J be the set of known customer jobs and V be the set of available chauffeur-company car pairs. As pre-announcements and the information they carry are the basis of this paper, we make an assumption that all jobs are pre-announced. The time line

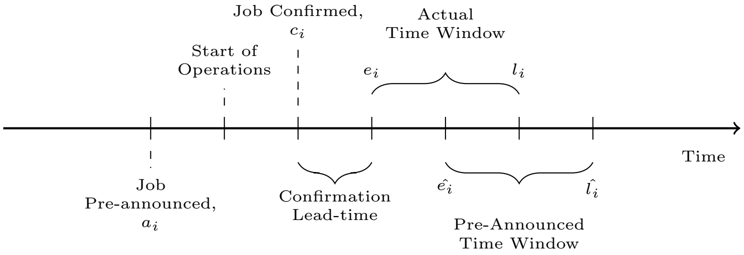


Figure 2: Time line of a pre-announced job.

of a job and its associated information events are shown in Figure 2. When a customer $i \in J$ preannounces their service request at time a_i before the start of operations, they state their pick-up location, drop-off location, and expected earliest pick-up time, $\hat{e}_i$. The locations are guaranteed to be accurate, but the pre-announced pick-up time may not be accurate. Thus, customer i subsequently calls at time c_i to confirm the job and announce their actual earliest pick-up time, e_i. Note, we assume that the confirmation time, c_i, always occurs a fixed, deterministic amount of time (the confirmation lead-time) prior to the actual earliest pick-up time, e_i. Furthermore, in this problem, we assume a fixed, deterministic length for all time windows. Thus, by knowing the pre-announced and actual earliest pick-up times, $\hat{e}_i$ and e_i, respectively, we can calculate the expected and actual latest pick-up times, $\hat{l}_i$ and l_i, respectively.

In our experiments, we vary both the number and accuracy of the inaccurate pre-announcements. We can adjust the level of accuracy by varying the difference between the pre-announced and actual time windows. If the pre-announced time window and the actual time window align, then the information is accurate; otherwise, it is inaccurate. Section 3 describes the specific nature of the instances that we test. Despite having knowledge of all job locations at the start of operations, the problem is still dynamic as the job confirmations arrive over time. To that end, we implement a rolling horizon framework. At each decision epoch, we solve a Mixed Integer Program (MIP) with the objective of minimizing costs while obeying all presumed time windows. The presumptions made about the time windows serve to distinguish the various routing strategies (see Subsection 2.3). However, all strategies rely on the use of the MIP: the topic of the following subsection.

2.1 Solving the Off-line Problem

We use a mixed integer program (MIP) from Yang et al. (1999) to formulate and solve the pick-up and delivery problem with time windows. The model was originally developed in the context of freight logistics as a method for rapid, online routing of vehicles in a truckload pick-up and delivery setting. The model, formulated as an assignment problem, seeks a set of least cost cycles that describe the order in which each company car should serve the jobs, the company cars that should remain idle, and the jobs that should be rejected.

The model of Yang et al. (1999) operates over two sets of nodes: one set of nodes, $V = \{1, \dots, K\}$, representing the company cars and one set of nodes, $J = \{1, \dots, N\}$, representing the customers. It is important to note that each customer node actually represents a revenue generating trip from a pick-up location to a drop-off location. Thus the distances between any two nodes are asymmetric.

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 $x_{i,j}$ representing the presence (1) or the absence (0) of an arc between the nodes in the union of the two sets, V and J; i.e. in the set: $V \cup J = \{1, \dots, K, K+1, \dots, K+N\}$. The timing variables are the continuous variables δ_i representing the time of arrival at the pick-up location of customer i.

The business goal of the designated driver service provider is to maximize the profit. This involves a trade-off between minimizing the routing costs of the service and maximizing the revenues of the served jobs. The MIP achieves this objective by minimizing 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 up front. 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 $l \in V$ is the cost (0, in our case) of allowing vehicle k to remain idle ; d^O_{ki} 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 pick-up 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 job i drop-off location to the pick-up 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^H_{ik} 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: $\min \sum_{k=1}^{K} \sum_{i=1}^{N} d^O_{ki} x_{k,K+i} + \sum_{i=1}^{N} \sum_{j=1}^{N} d_{ij} x_{K+i,K+j} + \sum_{i=1}^{N} \sum_{k=1}^{K} d^H_{ik} x_{K+i,k}$.

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 pick-up locations for the jobs. The routing related constraints 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 customer node is only feasible if it is time feasible, i.e. if it is possible to start service within the job's presumed time window ($\bar{e}_i \leq \delta_i \leq \bar{l}_i$) given the time that the company car becomes available and the time required to travel from the available location to the pick up location. We disallow any delay in arrival to the pick-up location. Similarly, 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.

For more detail on this formulation, the reader is directed to Yang et al. (1999) and Yang et al. (2004). The following subsection focuses on the use of the MIP in a rolling horizon framework.

2.2 Optimization in an Online Context

We incorporate the above model in a rolling horizon framework that runs the optimization each time a job is confirmed to the system, that is at every c_i. Figure 3 illustrates the interplay of the real-world 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 pre-announced) at time c_i excluding the jobs that were rejected earlier in execution. A job is rejected in execution only if it cannot be scheduled for service within its pick-up time window. A pre-announced job is never rejected before its confirmation time. We assume that the optimization run is instantaneous. This is not unrealistic as our off-line optimization problem is solved within one minute.

The route resulting from an optimization run is parsed according to the jobs and executed as required to meet the pre-announced or confirmed time windows. Specifically, in the optimization run at time $t = c_i$ we determine the latest time t_j^v that vehicle v should start moving to the pick-up location of job j to arrive in time ($t_j^v \geq t$). 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. If the pre-announced pick-up time passes without confirmation or another job confirms its pick-up time before job j is actually picked up, then vehicle v may be diverted to another job.

All jobs with $t_j^v > c_i$ are always included in the pool of unserved jobs that are part of the optimization run at c_i. 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. In short, if a chauffeur is not driving a customer's car, s/he is either moving in a company car to a customer's pick-up location, waiting in a company car at either the pick-up or the drop-off location of a customer, or waiting at the depot. If the chauffeur is moving, in a company car, to or waiting at the pick-up location of an unconfirmed pre-announcement, s/he may be re-routed to serve another job.

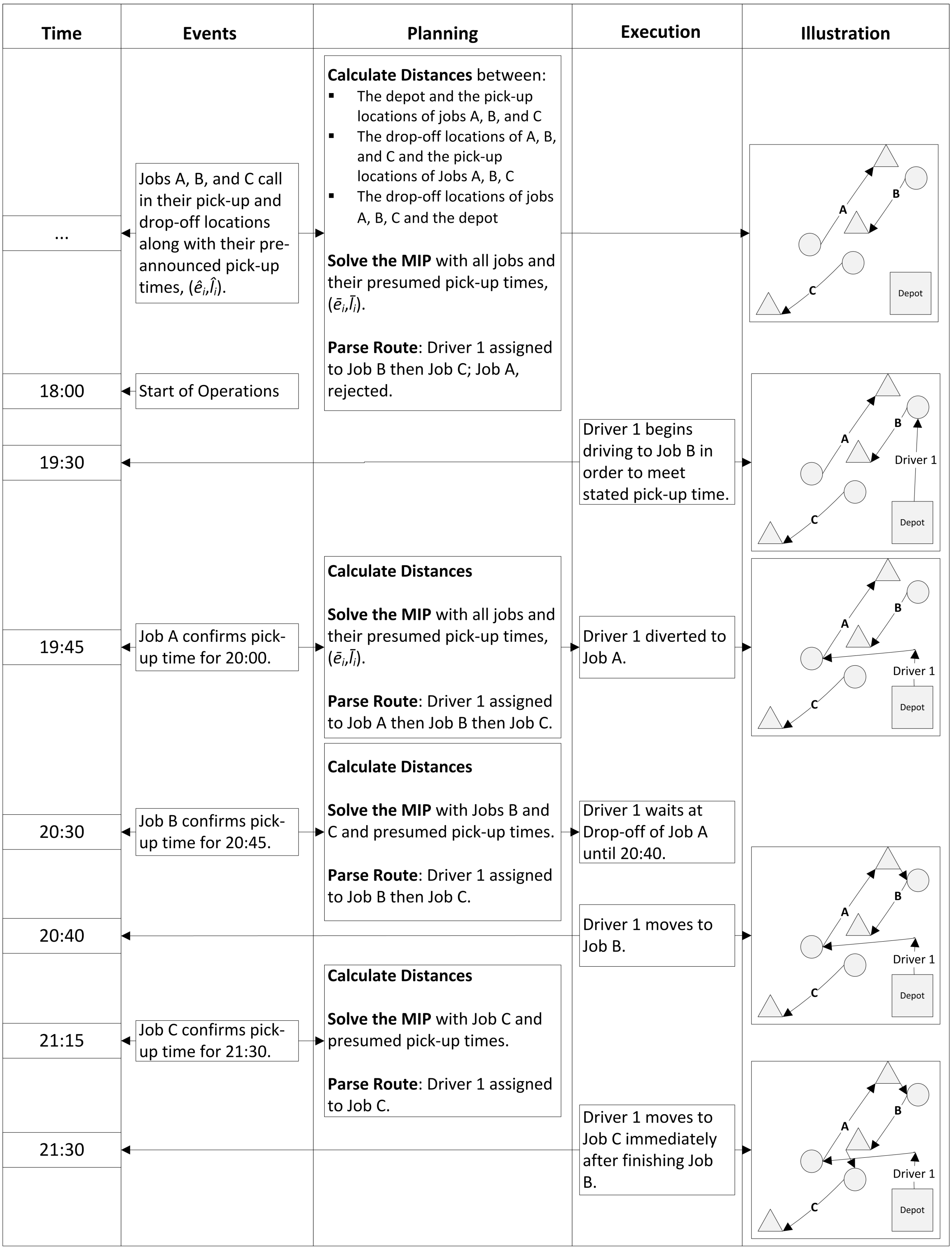
The manner by which we include the pre-announced jobs with their "presumed" time window information, $(\bar{e}_i, \bar{l}_i)$, defines our various routing strategies, the topic of the following subsection.

2.3 Routing Strategies

While the MIP remains at the core of all our strategies, the manner in which the time window information is fed to the MIP determines a strategy. Specifically, we term the time window information used by the MIP "the presumed time window information" and denote it as $\left(\bar{e}_i, \bar{l}_i\right)$. In this regard there are three primary strategies that we examine:

- *Ignore*: Ignore pre-announcements until they confirm at time c_i and we know their time window. Thus, job i is not included in the optimization model until $t \geq c_i$ at which point $(\bar{e}_i, \bar{l}_i) = (e_i, l_i)$.
- *Naïve*: Schedule pre-announcements as if the pre-announced 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 $t \geq c_i$ job i is included with $(\bar{e}_i, \bar{l}_i) = (e_i, l_i)$.
- *Multiple Time Scenarios* (*MTS*): Make different plans based on multiple potential time scenarios for pre-announcements, implement the most similar plan according to a function that considers the next jobs and the vehicle assignments. Thus at $t < c_i$, we include job i, but build multiple scenarios with $(\bar{e}_i, \bar{l}_i) = (\hat{e}_i \pm \mathcal{N}(0, \sigma^2), \bar{e}_i + s)$, where s represents the fixed, deterministic length of the time window; when $t \geq c_i$, we set $(\bar{e}_i, \bar{l}_i) = (e_i, l_i)$.

Moreover, we include an *off-line* benchmark, *A Priori*, in which all jobs are known in advance with 100% accurate time window information. Given the nuances of the *MTS* strategy, we devote the remainder of this section to describing the *MTS* strategy in detail.



Time	Events	Planning	Execution	Illustration
…	Jobs A, B, and C call in their pick-up and drop-off locations along with their pre-announced pick-up times, $(\hat{e}_i,\hat{l}_i)$.	**Calculate Distances** between: ▪ The depot and the pick-up locations of jobs A, B, and C ▪ The drop-off locations of A, B, and C and the pick-up locations of Jobs A, B, C ▪ The drop-off locations of jobs A, B, C and the depot **Solve the MIP** with all jobs and their presumed pick-up times, $(\bar{e}_i,\bar{l}_i)$. **Parse Route**: Driver 1 assigned to Job B then Job C; Job A, rejected.		
18:00	Start of Operations			
19:30			Driver 1 begins driving to Job B in order to meet stated pick-up time.	
19:45	Job A confirms pick-up time for 20:00.	**Calculate Distances** **Solve the MIP** with all jobs and their presumed pick-up times, $(\bar{e}_i,\bar{l}_i)$. **Parse Route**: Driver 1 assigned to Job A then Job B then Job C.	Driver 1 diverted to Job A.	
20:30	Job B confirms pick-up time for 20:45.	**Calculate Distances** **Solve the MIP** with Jobs B and C and presumed pick-up times. **Parse Route**: Driver 1 assigned to Job B then Job C.	Driver 1 waits at Drop-off of Job A until 20:40.	
20:40			Driver 1 moves to Job B.	
21:15	Job C confirms pick-up time for 21:30.	**Calculate Distances** **Solve the MIP** with Job C and presumed pick-up times. **Parse Route**: Driver 1 assigned to Job C.		
21:30			Driver 1 moves to Job C immediately after finishing Job B.	

Figure 3: The method by which the MIP is used as part of a rolling horizon framework.

2.3.1 The *MTS* Strategy

The Multiple Scenario Analysis (MSA) planning strategy was first introduced by Bent and Van Hentenryck (2004) for a vehicle routing problem with probabilistic information on job locations. In their original strategy, multiple job location scenarios were generated following known distributions from historical data. In our work, we do not plan for jobs whose locations are not yet revealed. Instead, we plan for jobs whose location is known, but the timing is not. Thus, we adapt the MSA strategy to generate multiple scenarios with potential time window realizations for all of the pre-announced jobs. We term our strategy the *Multiple Time Scenario* (*MTS*) strategy.

A scenario is created by adjusting each customer's pre-announced time window opening time. This adjustment yields a presumed time window, $(\bar{e}_i, \bar{l}_i)$, where $\bar{e}_i$ is obtained by adding to $\hat{e}_i$ a value that is drawn from a normal distribution with a mean of zero and a standard deviation equal to that found through an analysis of historical data, i.e. $\bar{e_i} = \hat{e_i} \pm \mathcal{N}(0, \sigma^2)$. The closing time, $\bar{l}_i$ is calculated by adding the length of the time window, s, to this opening time. If a value is drawn that leads to an earliest pick-up time, $\bar{e}_i$, that has already passed in the context of the simulation (i.e. is earlier than the current time plus the confirmation lead-time), then we draw a new value from the distribution.

For each scenario, a route plan is generated by solving the off-line problem as described in Subsection 2.1. Of the set of generated plans, Π, a distinguished plan π^* is selected. For Bent and Van Hentenryck (2004), the distinguished plan was the plan that was most similar to all others as judged by a consensus function. The main idea is that the most similar plan represents a least-commitment strategy (Stefik, 1981) and as such is most likely to accommodate any given future realization. The function that allows for this determination is a sum over the counts of vehicle to job pairs. The consensus function that we use is similar to Bent and Van Hentenryck (2004) as it compares different route plans based on the immediate, next job after the job in execution at different points in time. However, we assess the similarity on two dimensions separately: the route sequence and the job's vehicle assignment.

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 π 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 π, we can now define our consensus function for the *MTS* strategy as:

$$f_{MTS}(\pi) = \sum_{j \in J} \sum_{k \in V} y_{j,k,\pi} \left(Y_{j,k} + \sum_{v \in V} Y_{j,v} \right). \tag{1}$$

The first term, $y_{j,k,\pi} Y_{j,k}$, within the consensus function summation (1) 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 π. Our function extends this count by including a sum over v of $Y_{j,v}$ which allows us to consider the route sequence, independent of the vehicle assignment. This, in turn, places additional weight on the role of the immediate, next job in a route sequence, which is especially important in the context of the pick-up and delivery problem given its asymmetric distances. This 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.

We select as our distinguished plan $\pi^* \in \Pi$ such that $f(\pi^*) \geq f(\pi)$ for every $\pi \in \Pi$. We break ties arbitrarily by selecting the first of the highest scoring plans.

To illustrate the difference between our consensus function and that of Bent and Van Hentenryck (2004), consider an example with two chauffeurs and five customers for which three plans were generated. The first plan assigns customers five, three, and one to chauffeur one (in that order) and customers two and four to chauffeur two (in that order). The second plan assigns customers two and four to chauffeur one (in that order) and customers five, three, and one to chauffeur two (in that order). In Plan 3, chauffeur one serves customers five, two, and one (in that order) and chauffeur two serves customers three and four (in that order). These three plans are summarized in Table 1.

Following the consensus function of Bent and Van Hentenryck (2004), Plans 1 and 3 each score three points (two points each for customer five as the first job for chauffeur one and one point for customer two as the first job for chauffeur two [in Plan 1] and one point for customer three as the first job for chauffeur two [in Plan 3]) and Plan 2 scores two

Table 1: Example of a scoring plans MSA versus *MTS*.

	Chauffeur		Plan Score	
	1	2	f_{MSA}	f_{MTS}
Plan 1	5, 3, 1	2, 4	3	8
Plan 2	2, 4	5, 3, 1	2	7
Plan 3	5, 2, 1	3, 4	3	7

points (one point for job two as the first job for chauffeur one and one point for job five as the first job for chauffeur two). This implies that either Plan 1 or Plan 3 is selected as the distinguished plan. Following our function, Plan 1 scores eight points (five points, for the first position and assignment to chauffeur one of job five and three points for the first position of and assignment of job two to chauffeur two) and Plans 2 and 3 tie with seven points each. (Plan 2 receives three points for job two as the immediate, next job with chauffeur one, and four points for job five assigned to chauffeur two; while Plan 3 receives five points for job five as the immediate, next assigned to chauffeur one and two points for job three as the immediate, next with chauffeur two.) This implies that Plan 1 is selected as the distinguished plan.

In this example, if the vehicles are identical then Plan 1 and Plan 2 are identical in terms of their routes. Thus, it is reasonable to expect that one of the two "identical" plans would be selected by the consensus function. Our function captures this by giving more weight to the fact that customer five is the first job in all plans even if not with the same chauffeur. The MSA consensus function, on the other hand, downgrades Plan 2 on the basis that job five is assigned to chauffeur two; this has the effect of causing a tie between the clearly distinct Plans 1 and 3.

Each of the information strategies (*Ignore*, *Naïve*, and *MTS*) described here was tested on a set of operational instances designed to mirror the structure and timings of jobs encountered by an actual designated driver service in the Netherlands; the construction of these instances is the topic of the next section.

3 Computational Experiments

To compare the different strategies, we generated 30 test instances based on data from a designated driver service in Rotterdam, The Netherlands. The cost to the designated driver service is .30 euros/km for each kilometer driven by a company car; the revenue for each job includes a fixed fee of 6 euros plus 3 euros/km traveled in the customer's vehicle. Each instance consists of three chauffeurs and a set of 20 jobs; in practice, this number of jobs is consistent with a typical evening of operations. Each job has a given origin, destination, announcement time, confirmation time, and two time windows: pre-announced and real. The parameter settings associated with each of these pieces of information define our test cases.

We distinguish between jobs that originate from the city center (in) and from outside the city center (out). The proportion of jobs going from in to out were apportioned based on a normal distribution with mean of .6 and standard deviation of .1; the jobs not going from in to out were apportioned to an out to in configuration based on a mean of .6 and standard deviation of .1. The remaining jobs follow an out to out configuration. The average job distance is 31 km, with a minimum distance of 10 km and a maximum distance of 125 km. See Figure 4 for an example instance where customer origins are marked by a circle and the corresponding destination by a linked triangle. The distances between pick-up and drop-off points, between jobs, and between jobs and the depot are calculated as straight line distances. These straight line distances are then multiplied by 1.54 to reflect the actual distance experienced over the road network of the Netherlands. Finally, when converting between road distance and travel time a multiplier of 1.2 is used to reflect the average speed on the roads used; all travel times are assumed to be deterministic. The multipliers used to convert from straight line distance to travel times were derived empirically from Google maps data for the Rotterdam area.

All jobs are pre-announced; thus, the job locations are known at the start of operations. The time window for pick-up has a length of 30 minutes; the earliest pick-up time is 15 minutes after the confirmation time and the latest pick-up time is 30 minutes later or 45 minutes after the confirmation time. Job confirmation times are drawn from a normal distribution with a mean of 10:30pm and a standard deviation of 2 hours. Any jobs that randomly come up before the start of operations at 6pm are mapped to the start of operations. Based on the same 30 instances, we generated cases with different numbers of inaccurate pre-announcements (25% [5 jobs], 50% [10 jobs], 75% [15 jobs], and 100% [20 jobs]).

All jobs provide an estimate of their service time window prior to the start of operations. An accurate pre-announced time implies that the pre-announced time, $(\hat{e}_i, \hat{l}_i)$, is equal to the actual time, (e_i, l_i). For inaccurate jobs, however, the actual time window deviates from the pre-announced estimate. To investigate the impact of the inaccuracy of the pre-announced information, but allow for comparisons between the different cases, we

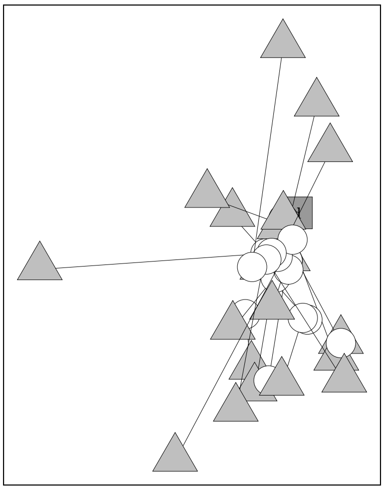


Figure 4: Example instance with customer origins (circle) and destinations (triangle)

fix the actual service time windows and generate different pre-announced time window estimates. We determine the pre-announced time windows by taking the start of the actual time window and adding to it a value drawn from a truncated normal distribution. In the base case, which most closely resembles our practical setting, the distribution has a mean of 0 and a standard deviation of 30 minutes. During the construction of these instances, we did not allow values outside the range from -60 to 60. Any values that fell outside of the range were re-drawn from the allowable range. As such, the pre-announced times follow a truncated normal distribution, with dispersion (measured by standard deviation) equal to the desired amount of offset in the pre-announced time. Thus, when we say an average deviation of 30 minutes, we mean that the start of the pre-announced time is most likely (68% probability) to fall within 30 minutes before or 30 minutes after the actual start of the time window.

In the experiments, where the *MTS* strategy is used, we choose to generate 30 scenarios. The choice of 30 scenarios was driven by a trade-off between time and quality. When using the *MTS* strategy to generate a plan, the MIP must be run for as many scenarios as we generate. That is, if we use 30 scenarios we must solve the MIP 30 times in one planning phase. While this can be done efficiently for the current problem size, with larger instances we found that using more scenarios violated our assumption of instantaneous route planning. As we are using this strategy as part of a rolling horizon framework, speed is essential.

4 Results

We implemented the simulation in C++ using Gurobi 5.0.1 as the MIP solver (Gurobi Optimization, 2012) and ran all instances on an Intel i5 2.4 GHz computer. The value of information accuracy is judged here by comparing the routing and rejection costs, averaged across 30 days of simulated operations, for each strategy. We begin, in this section, by examining the value of knowing whether the pre-announced pick-up time is accurate or not. We then continue to examine the value of knowing how inaccurate the pre-announced time window might be on average. We conclude by studying the value of knowing the direction (early/late) of the inaccuracy.

4.1 Knowing which Pre-announced Times are Accurate

First, we study the value of knowing, in advance, which pre-announcement times are accurate or not by comparing two information contexts. In the first context, the "uninformed" context, we assume that we do *not* know whether the pre-announced pick-up time, as specified by the customer, is accurate or not. This means that the simulated dispatching system knows, at the start of operations, the pre-announced job locations and time windows, but does not know if the time window information is accurate or not.

In the second information context, the "informed" context, we assume to know in advance, at the start of operations whether the pre-announced time window is accurate or not, for each job. The "informed" context mirrors reality to some extent as dispatchers often have a sense of which customers are providing accurate information or not. For example, if a customer is seeking a ride home from a football match at a stadium, then the estimate of the pick-up time is probably more accurate than if a customer is seeking a ride home from a night out with friends.

For each information context, we consider varying percentages of inaccurate pre-announcements, which are, on average, inaccurate by an average deviation of 30 minutes from the real pick-up time. We apply the three different strategies and the *A Priori* benchmark to the two information contexts. In the uninformed context, the *MTS* strategy always applies the scenario generation mechanism to handle all time window information. In the informed context, the *MTS* strategy directly uses the pre-announced time windows when they are known to be accurate, but applies the scenario generation mechanism to the pre-announced pick-up times that are known to be inaccurate. To distinguish between the two information contexts, we denote the use of the *MTS* strategy in the informed context as *MTS - I*.

In the informed context, we also examine an additional strategy, *Ignore - I*, which uses the pre-announced time window information for the accurate jobs while ignoring, until confirmed, the inaccurate jobs. This strategy may be seen as a hybrid between the *Naïve* and the *Ignore* strategy: if all jobs are accurate, the strategy is equivalent to the *Naïve* strategy and if all the jobs are inaccurate, it is equivalent to the reactive strategy of *Ignore*.

Table 2 summarizes the average total costs (over 30 instances) along with the standard error of these costs for various percentages of inaccurate jobs. From the table, we see that there is a nearly 44% gap in costs between the *A Priori* benchmark and the *Ignore* strategy, which represents the value of perfect advance information. We see that knowing which pre-announced times are accurate is valuable: *MTS-I* performs an average of 9% better in terms of total costs across all percentages of inaccurate jobs relative to *MTS*. We observe a similar outcome for the *Ignore* strategy as the *Ignore - I* strategy outperforms the purely reactive *Ignore* strategy.

In the informed context, we see that incorporating the inaccurate jobs in the route plan in a stochastic manner provides gains and the *MTS - I* strategy consistently outperforms all other strategies. However, we also see that simply ignoring the inaccurate jobs is not a good strategy. The hybrid, *Ignore - I* strategy does worse than the *Naïve* strategy across all percentages of inaccurate jobs. This suggest that it is better to use the inaccurate information as if it were accurate than to ignore it.

We also see that the *Naïve* strategy outperforms the *MTS* strategy when the percentage of inaccurate jobs is low, i.e. 50% inaccurate jobs or less. This is due to the fact that the *MTS* strategy was designed to cope with inaccuracy by generating a set of scenarios for each proposed time window, but when accuracy prevails, these perturbations of reality come at a cost. However, when 100% of the jobs are inaccurately pre-announced, the *MTS* strategy outperforms the *Naïve* strategy, in terms of total costs, by 7% (a statistically significant result with $p = .03$ in a paired t-test).

Figure 5 illustrates the differentiating features of these strategies across the two information contexts. The figure highlights the average number of empty kilometers, the cost of rejecting jobs, the total costs, and the profits associated with each routing strategy across the varying levels of inaccuracy. Overall, we observe that the performance of the system deteriorates with the percentage of inaccurate jobs. The decline in performance is roughly proportional to the number of inaccurate jobs.

We note in Figure 5(a) that the routing costs as measured by the empty kilometers traveled for the *Ignore* strategy is the lowest. However, this lower travel distance comes

Table 2: Average ± standard error of total costs for six strategies at 30 minutes of deviation with five different percentages of jobs with inaccurately pre-announced pick-up times; $n = 30$.

Number of			Uninformed			Informed	
Inaccurate Jobs		**A Priori**	**Ignore**	**Naïve**	**MTS**	**Ignore-I**	**MTS-I**
0	[0%]	513.6 ± 26.5	911.8 ± 40.8	513.6 ± 26.5	566.5 ± 28.9	513.6 ± 26.5	513.6 ± 26.5
5	[25%]	513.6 ± 26.5	911.8 ± 40.8	582.0 ± 28.7	605.9 ± 30.6	586.8 ± 31.3	567.4 ± 29.9
10	[50%]	513.6 ± 26.5	911.8 ± 40.8	620.2 ± 31.0	627.3 ± 32.7	669.6 ± 29.2	585.4 ± 30.9
15	[75%]	513.6 ± 26.5	911.8 ± 40.8	636.2 ± 29.1	623.2 ± 28.9	832.0 ± 37.5	621.3 ± 26.3
20	[100%]	513.6 ± 26.5	911.8 ± 40.8	701.2 ± 32.3	648.9 ± 31.0	911.8 ± 40.7	648.9 ± 31.0

at the expense of higher job rejection costs. This trade-off in costs stems from the fact that the confirmation lead time combined with the length of the time window may not, in some cases, be long enough for the chauffeur to travel to the customer in time. Thus, more jobs are rejected, while accruing fewer empty kilometers. Intuitively, we see that using the advance information on the accurate jobs helps *Ignore - I* to decrease its rejection costs relative to *Ignore*, especially when the percentage of inaccurate jobs is low.

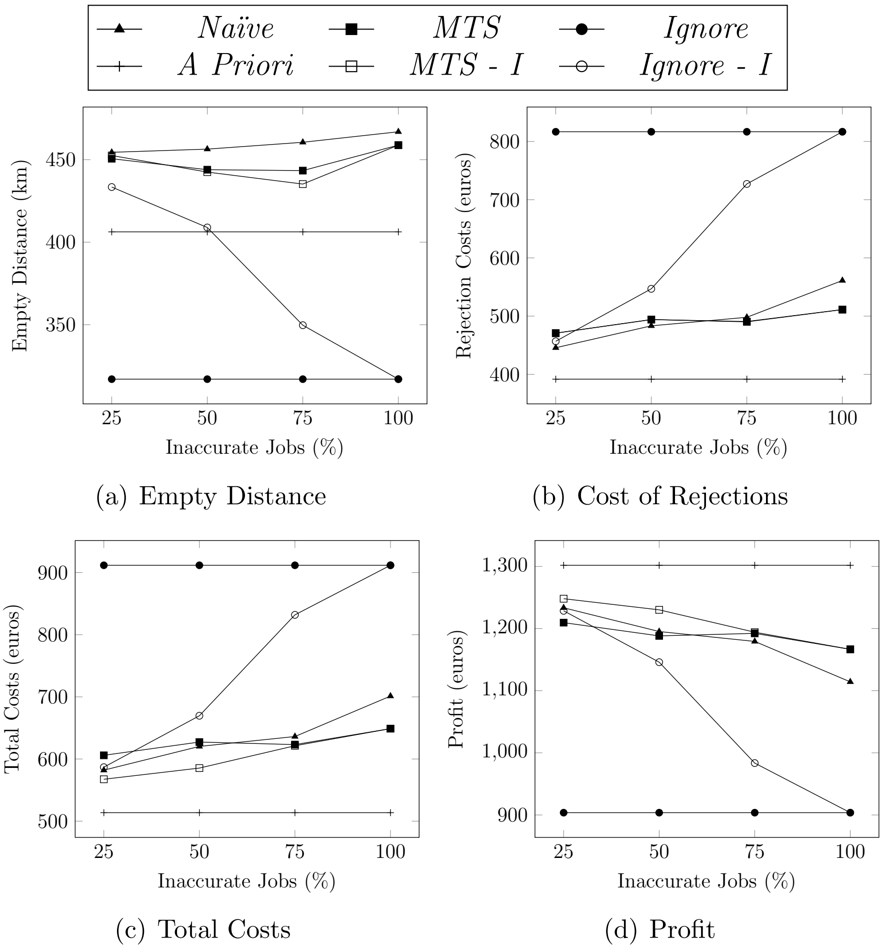
Furthermore, by examining the performance of the *MTS* strategy, we see that it is the comparatively lower rejection costs that give this strategy the advantage, both when the accuracy of the pre-announced time windows is and isn't known. Similarly, the *Naïve* strategy incurs more empty kilometers than the other strategies, but has relatively lower job rejection costs. In contrast to the *Ignore* strategy, this tradeoff in costs arises from the fact that the *MTS* and *Naïve* strategies may move towards a particular pickup location before the customer actually confirms his/her actual service time. While this may prevent a job rejection, it may also require an en route diversion as other, more urgent, jobs are confirmed.

In contrast to, but consistent with, the total costs, we depict the average profits for the different percentages of inaccurate jobs. As expected, we see that the profit deteriorates with the increase in the number of inaccurately pre-announced jobs, especially for the simpler *Ignore* and *Naïve* strategies. Overall, we observe that the *MTS* strategy is very robust. *MTS* is the best strategy for both information contexts. Comparing the two contexts, *MTS - I* outperforms *MTS* by 3% in terms of profit (1230.12 versus 1188.19) and by 7% in terms of cost (585.35 versus 627.29), when 50% of the jobs are inaccurate. This 3% gain in profit or 7% decrease in costs represents the value of the additional information on accuracy.

From these results, we can say that if the degree of inaccuracy is limited (30 minutes of deviation, on average) then knowing whether a client is accurately forecasting their pick-up time is beneficial when using a routing strategy that can incorporate the inaccurate jobs in a stochastic manner rather than ignoring them. Whether this holds true when the inaccuracy of the pre-announced time window is greater than 30 minutes of deviation is the topic of the next section.

4.2 Knowing the Degree of Inaccuracy

We now study the gains associated with knowing something about the degree of inaccuracy. Specifically, we examine the informed *MTS-I* strategy relative to one strategy in



(a) Empty Distance

(b) Cost of Rejections

(c) Total Costs

(d) Profit

Figure 5: Impact of different percentages of inaccurate jobs (20 jobs total) on empty distance and job rejections of routing strategies at 30minutes of deviation and two different levels of information on accuracy; $n = 30$.

which the dispersion is underestimated and one in which the dispersion is overestimated. The application of the *MTS* strategy in the context of underestimation is termed *MTS - UE* and in the context of overestimation is termed *MTS - OE*. In these experiments, for the underestimated context we use a standard deviation of 15 minutes less that the tested degree of inaccuracy and for the overestimated context a standard deviation of 15 minutes greater than the tested degree of inaccuracy. It is important to study the performance of a poorly calibrated system, as it is often the case that estimations of stochastic parameters made from past data do not necessarily match future operations exactly. We compare these strategies across varying degrees of inaccuracy (30, 45, 60, 75 and 90 minutes of average deviation) when 100% of the jobs are inaccurately pre-announced.

To generate the instances with higher degrees of inaccuracy, but still allow for comparisons across the different degrees of inaccuracy, we use the instances in the base case (30 minutes of deviation) to build the new instances. In the base instances, we first identify the direction (early or late) of the deviation of the pre-announced time window opening relative to the actual time window opening time. We then, use the pre-announced time window opening in the base case and subtract (if early) or add (if late) an amount equal to the difference of the desired deviation and the base case 30 minutes of deviation. For example, if we wish to construct an instance with a 60 minute deviation, on average, then we will subtract or add 30 minutes of deviation to each pre-announced time window opening in the base case. In this way, we created cases with 45, 60, 75, and 90 minutes of deviation.

Table 3 shows the results for a set of experiments in which we vary the expected degree of inaccuracy when all jobs are inaccurate. As expected, we observe that the performance of the system deteriorates with the degree of inaccuracy. This is especially true for the *Naïve* strategy that shows an increase in total costs of 35% when comparing the performance at 90 minutes to the performance at 30 minutes of expected inaccuracy. With only a 12% increase in costs the *MTS-I* is much more robust and continues to outperform both the *Naïve* strategy and the *Ignore* strategy (total costs as in Table 2: 911.8 ± 40.8). Interestingly, when the pre-announced time window information deviates, on average, by 90 minutes, the *Ignore* strategy outperforms the *Naïve* strategy. Thus, when the degree of information inaccuracy is very high, a fully dynamic, reactive strategy is better than a strategy that blindly uses the timing information as given. This result is interesting as it serves to highlight the trade-off between the use of inaccurate information given during planning and accurate information arriving during operations.

When considering the value of a perfectly calibrated stochastic strategy, we find that

Table 3: Average ± standard error of total costs of four strategies across five different degrees of inaccuracy with 100 percent inaccurate jobs; $n = 30$.

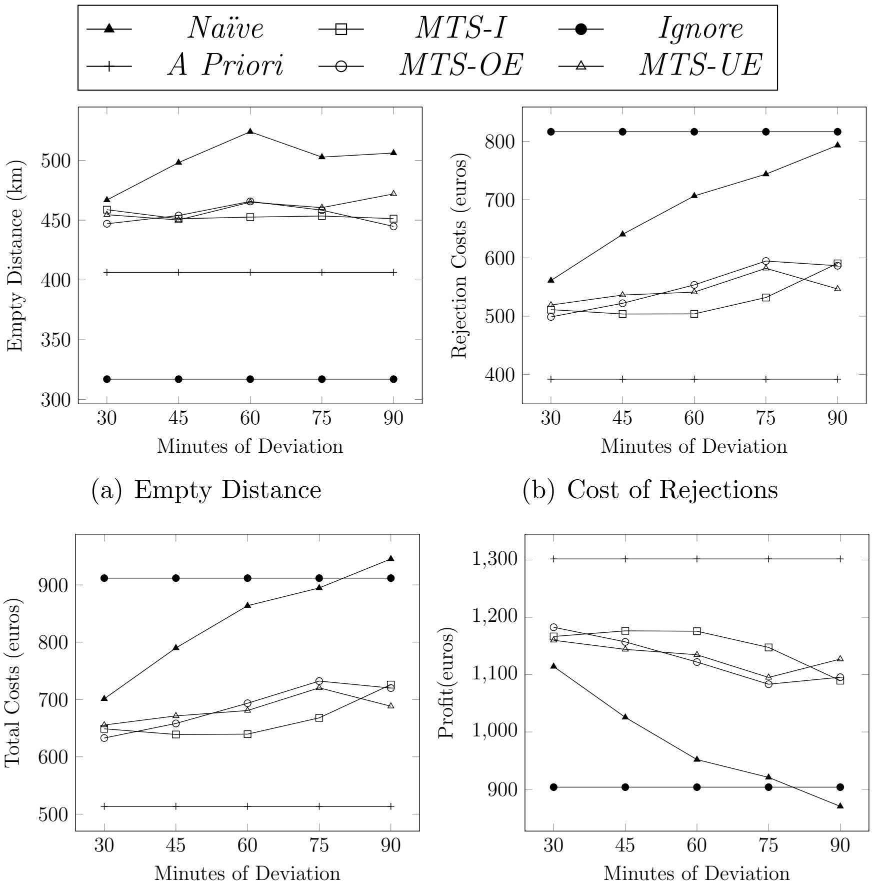
Degree of Inaccuracy	**Naïve**	**MTS-I**	**MTS - UE**	**MTS - OE**
30 min	701.2 ± 32.3	648.9 ± 31.0	655.5 ± 30.8	632.8 ± 30.5
45 min	790.0 ± 36.8	679.3 ± 32.3	671.4 ± 35.1	658.2 ± 35.2
60 min	863.7 ± 35.5	677.0 ± 30.0	680.8 ± 36.4	693.5 ± 36.0
75 min	894.7 ± 36.8	668.0 ± 31.0	720.4 ± 37.8	732.2 ± 38.6
90 min	945.3 ± 31.9	726.0 ± 34.7	688.3 ± 34.8	720.0 ± 35.1

the results are interesting as overestimating the actual deviation (*MTS - OE*) is only beneficial relative to underestimating the deviation (*MTS - UE*) when the degree of inaccuracy is 45 minutes or less. When the degree of inaccuracy is 60 minutes or greater, then it is better to underestimate the inaccuracy. In Figure 6, we can examine this phenomenon closer by studying the cost components of the slightly perturbed *MTS* strategies, *MTS-OE* and *MTS-UE*. We find that they both perform similarly to *MTS-I* in terms of routing costs, but incur higher rejection costs when the degree of deviation is moderate (between 45 and 70 minutes). Interestingly, when the degree of deviation is low (30 minutes) then overestimating the degree of deviation is the better strategy – likely because the additional slack in the schedule allows for fewer job rejections relative to *MTS-I*. On the other hand, when the degree of deviation is high (90 minutes) then underestimating the degree of deviation is better – likely because the vehicles are forced into the field earlier and can capture more jobs than the *MTS-I* strategy.

4.3 Knowing the Direction of Inaccuracy

We now consider the value of knowing the direction of the inaccuracy, i.e. early or late. This information condition represents the context in which an experienced dispatcher may know that certain customers have a tendency to confirm earlier or later than their pre-announced time. Therefore, we assume that for each pre-announcement, we know whether the customer will actually want to be served earlier or later than the pre-announced time. Given this information, we then draw the different time scenarios from only one side of a normal distribution around the pre-announced time; we call this application of the *MTS* strategy *MTS - E/L*.

In Table 4, we present the results of the *MTS* strategy applied in three different infor-



(a) Empty Distance

(b) Cost of Rejections

(c) Total Costs

(d) Profit

Figure 6: Impact of different degrees of inaccuracy on profit of six strategies when all jobs are inaccurate; $n = 30$.

Table 4: Average ± standard error of total costs of three different strategies across four percentages of inaccurate jobs with 30 minutes of deviation; $n = 30$.

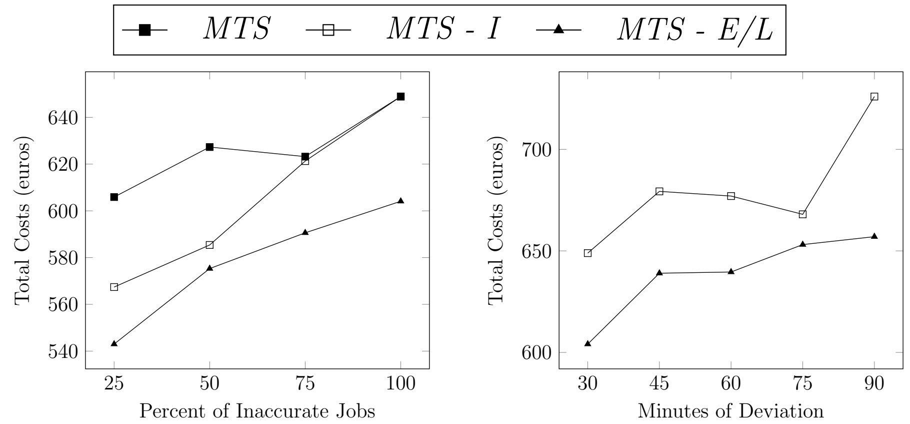
Number of Inaccurate Jobs	**MTS**	**MTS-I**	**MTS - E/L**
5 [25%]	605.9 ± 30.6	567.4 ± 29.9	543.0 ± 27.5
10 [50%]	627.3 ± 32.7	585.4 ± 30.9	575.3 ± 28.2
15 [75%]	623.2 ± 28.9	621.3 ± 26.3	590.6 ± 27.1
20 [100%]	648.9 ± 31.0	648.9 ± 31.0	604.1 ± 28.0

mation contexts (uninformed, informed, and early/late) across four different percentages of inaccurate jobs (inaccurate by an average of 30 minutes). We see that knowing the direction of the inaccuracy (*MTS - E/L*) provides moderate gains over knowing only whether the pre-announced time is accurate or not (*MTS - I*). The fact that the differences are relatively small demonstrates that the *MTS* strategy works well even without any information on the accuracy of the pre-announced time windows. This is an important practical consideration as a planner may not have any information on the deviation of the pre-announced time windows.

Figure 7(a) illustrates these results showing that across the various percentages of inaccurate jobs, knowledge on which jobs are making early or late pre-announcements yields the lowest routing costs. Figure 7(b) highlights the capabilities of *MTS-E/L* relative to *MTS-I* across five degrees of deviation. Note that when varying the degree of the deviation, we only consider cases with 100% inaccurate pre-announcements, thus the uniformed *MTS* strategy is indistinguishable from its informed counterpart, *MTS-I.* In Figure 7(b), we find that across all degrees of inaccuracy, the early/late context yields the lowest costs relative to *MTS-I.* Thus, when the average degree of inaccuracy is high, the only information that a planner should seek is that on whether the pre-announced earliest pick-up time is early or late relative to the actual earliest pick-up time. The greatest advantage of knowing whether a pre-announcement is an early or late estimate comes when there is an average of 45 minutes or more of deviation.

5 Discussion

We open the discussion by returning to the statement of Psaraftis (1995) that advance information is often imprecise and accuracy only increases during operations. In the context studied here, this was certainly true – only when jobs are confirmed do we know



(a) MTS strategies at different percentages of inaccurate jobs (30 minutes of deviation).

(b) MTS strategies at different degrees of inaccuracy (100% of jobs inaccurate).

Figure 7: Average total costs for *MTS* operating with various levels of information across different percentages of inaccurate jobs (30 minutes of deviation) and different degrees of inaccuracy (100% inaccurate jobs); $n = 30$.

the precise time window information. This raised the question of when does advance inaccurate time window information provide value in the context of a dynamic pick-up and delivery problem with time windows?

Based on experiments with real-life data, we show that advance information, even if inaccurate, can provide benefits from a planning perspective. In settings with only moderate levels of inaccuracy (30 minutes deviation on average, in our context), the *Naïve* strategy, which treats the inaccurate information as accurate, outperforms the *Ignore* strategy, which ignores inaccurate information until confirmed. Only when the degree of inaccuracy is large (90 minutes deviation on average or more, in our context) is it better to employ the purely reactive *Ignore* strategy. The stochastic mechanism we introduced, the *MTS* strategy, is consistently robust – it outperforms both the *Naïve* and the *Ignore* strategies, in all cases.

Of course, the world is more intricate than that assumed by specifying a continuum from early, inaccurate to late, accurate information. By examining the *MTS* strategy across a spectrum of information types, we see that, when only 50% of the jobs are inaccurate, knowing which jobs are inaccurate gives the *MTS - I* strategy a 7% advantage in terms of costs over the *MTS* strategy when accuracy is unknown. Thus, even a simple indication of accuracy in the pre-announced time window information is worth a benefit in terms of routing costs. Furthermore, knowing whether a pre-announcement is earlier or later than the actual time window yields the best performance of the *MTS* strategy across all levels and degrees of inaccuracy.

The quantification of information in terms of routing costs allows for the consideration of alternative operating policies. For example, a company could offer a discount to customers who accurately pre-announce their pick-up times or leverage a penalty to customers who are very late relative to their pre-announced pick-up time. Additionally, these results indicate the value of a well calibrated stochastic routing system by highlighting the cost associated with over- and underestimating the degree of inaccuracy. Such results are important as more and more companies are seeking to mine large operational databases for the purpose of improved decision making.

As this is one of the first papers that specifically considers temporal uncertainty in the context of pickup and delivery problems, we see many opportunities for further research. The function for selecting the candidate plan in the *MTS* strategy could be refined further to accommodate the asymmetric nature of distances in a pick-up and delivery problem by including the full route of jobs for each vehicle or by studying specific pairings or strings of jobs. When considering the generation of scenarios, we could also extend the MTS strategy for use in situations where the pick-up is known, but the delivery location is not known (or known with some inaccuracy). Considering other contexts, we might also be able to apply the current *MTS* strategy to the problem of using harbor pilots to bring ships to berth which is another pick-up and delivery problem with known locations, but potentially inaccurately pre-announced time windows. Another application area is the transport of patients after hospital visits of uncertain lengths. In general, we see great value in studying the incorporation of stochastic time window information in optimization.

References

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`.

Dror, Moshe, Gilbert Laporte, Pierre Trudeau. 1989. Vehicle routing with stochastic demands: Properties and solution frameworks. *Transportation science* **23**(3) 166–176.

ETSC. 2008. Drink driving fact sheet. Tech. rep., European Transport Safety Council. URL `http://www.etsc.eu/documents/Fact_Sheet_DD.pdf`.

Gendreau, Michel, Jean-Yves Potvin. 1998. *Dynamic vehicle routing and dispatching*. Springer.

Ghiani, Gianpaolo, Francesca Guerriero, Gilbert Laporte, Roberto Musmanno. 2003. Real-time vehicle routing: Solution concepts, algorithms and parallel computing strategies. *European Journal of Operational Research* **151**(1) 1–11.

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.

Gurobi Optimization, Inc. 2012. Gurobi optimizer reference manual. URL `http://www.gurobi.com`.

Hyytiä, 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.

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.

NHTSA. 2012. Alcohol impaired driving, traffic safety facts: 2011 data. Tech. rep., National Highway Traffic Safety Administration. URL `http://www-nrd.nhtsa.dot.gov/Pubs/811700.pdf`.

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. 2012. A review of dynamic vehicle routing problems. *European Journal of Operational Research* **225**(1) 1–11.

Psaraftis, Harilaos N. 1995. Dynamic vehicle routing : Status and prospects. *Annals of Operations Research* **61** 143–164.

Sang-Hun, Choe. 2007. Drinkers in korea dial for designated drivers. URL `http://www.nytimes.com/2007/07/10/world/asia/10korea.html?_r=0`.

Savelsbergh, M.W.P., M. Sol. 1995. The general pickup and delivery problem. *Transportation science* **29**(1) 17–29.

Schilde, M., K.F. Doerner, R.F. Hartl. 2011. Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & Operations Research* **38**(12) 1719–1730.

Stefik, Mark. 1981. Planning with constraints (molgen: Part 1). *Artificial intelligence* **16**(2) 111–139.

Thomas, Barrett W. 2010. Dynamic vehicle routing. *Wiley Encyclopedia of Operations Research and Management Science*. Wiley Online Library.

Thomas, Barrett W., Chelsea C. White. 2004. Anticipatory Route Selection. *Transportation Science* **38**(4) 473–487.

Tjokroamidjojo, D., E. Kutanoglu, G. D. Taylor. 2006. Quantifying the value of advance load information in truckload trucking. *Transportation Research Part E* **42** 340–357.

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.

ERIM Report Series *Research in Management*	
ERIM Report Series reference number	ERS–2014–002–LIS
Date of publication	2014–01–27
Version	27–01–2014
Number of pages	30
Persistent URL for paper	`http://hdl.handle.net/1765/50394`
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: info@erim.eur.nl Internet: `http://www.erim.eur.nl`
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)