TRANSPORTATION SCIENCE
Vol. 44, No. 2, May 2010, pp. 193–205
ISSN 0041-1655 | EISSN 1526-5447 | 10 | 4402 | 0193

informs®

DOI 10.1287/trsc.1090.0303

A Model and Algorithm for the Courier Delivery Problem with Uncertainty

Ilgaz Sungur, Yingtao Ren, Fernando Ordóñez, Maged Dessouky
Industrial and Systems Engineering, University of Southern California, Los Angeles, California 90089
{sungur@usc.edu, yingtaor@usc.edu, fordon@usc.edu, maged@usc.edu}

Hongsheng Zhong
UPS, Timonium, Maryland 21093, hzhong@ups.com

We consider the courier delivery problem (CDP), a variant of the vehicle routing problem with time windows (VRPTW) in which customers appear probabilistically and their service times are uncertain. We use scenario-based stochastic programming with recourse to model the uncertainty in customers and robust optimization for the uncertainty in service times. Our proposed model generates a master plan and daily schedules by maximizing the coverage of customers and the similarity of routes in each scenario, while minimizing the total time spent by the couriers and the total earliness and lateness penalty. To solve large-scale problem instances, we develop an insertion-based solution heuristic, called master and daily scheduler (MADS), and a tabu search improvement procedure. The computational results show that our heuristic improves the similarity of routes and the lateness penalty at the expense of increased total time spent when compared to a solution of independently scheduling each day. Our experimental results also show improvements over current industry practice on two real-world data sets.

Key words: vehicle routing; robust optimization; stochastic optimization; insertion; tabu search
History: Received February 2008; revisions received: July 2008, April 2009, August 2009, September 2009; accepted: September 2009. Published online in *Articles in Advance* January 22, 2010.

1. Introduction

In this study, we consider the courier delivery problem (CDP), a variant of the vehicle routing problem with time windows (VRPTW) with uncertain service times and probabilistic customers. This problem is motivated by the operations of a courier delivery/pickup company that serves a dense urban area. In this situation, travel times between locations are relatively short, and therefore can be assumed constant when compared to the variation in service times at each location. In a business district, for example, a driver might have several drops and pickups in multiple offices at the same address. We therefore consider a routing problem with uncertainty because of unknown service times and the probabilistic nature of the customers, i.e., daily delivery requests from potential customers are not known beforehand, but they usually become available in the morning.

For many practical reasons, it seems beneficial to create regular or consistent routes for the CDP that assign the same driver to the same set of customers to serve them at roughly the same time. Such consistent routes are easy to adapt to the realization of the daily uncertainty and help courier companies realize the important goal of personalization of services, making the driver the contact person whenever the customer needs service. This regularity in schedules also increases driver familiarity with their own routes and territories, which improves driver efficiency.

A simple solution for the CDP, referred to as independent daily schedules, is to solve the routing problem each day based on that day's requirements, when the customer locations and service times are revealed at the beginning of the day (Savelsbergh and Goetschalckx 1995; Beasley and Christofides 1997). However, because there is no consideration of geographical area or regularity of service in such solution methods, it may not provide the level of consistency or regularity that real practice needs (Malandraki et al. 2001). One method used to obtain consistent routes is *territory planning*, which assigns service territories to drivers over a certain planning horizon (Zhong, Hall, and Dessouky 2007). The variations in demand are accommodated by adjusting the border of the territories. However, adjusting driver territories efficiently is not trivial and usually only a limited number of drivers share their capacities through territory adjustments. Recently (Groër, Golden, and Wasil 2008) introduce the Consistent VRP (ConVRP), and the objective is finding routes in which the same drivers visit the same customers at roughly the same time on each day the customers need service. This differs from the CDP in that matching drivers to customers is not an explicit objective of CDP and ConVRP does not consider time window constraints or uncertain service times.

1.1. Scientific Contribution

In this work, we present a mixed-integer program (MIP) model that aims to obtain a master route or plan and develop an iterative heuristic solution method for it. This master route is then used as a basis to construct daily schedules for couriers to meet the delivery requests at minimum cost. Thus the master plan eliminates significant replanning in each day and increases the similarity between daily schedules and the familiarity of drivers with the daily routes, thereby potentially improving driver productivity.

Compared with territory planning, the master plan model should allow additional flexibility in building daily schedules as more drivers can share their capacities in the daily adjustment. Likewise, the master plan model also should be more flexible in constructing daily routes than ConVRP, as there is no explicit objective to match drivers to customers. We compare the performance of the master plan method with existing methods in the computational section of this paper.

To formulate the CDP, we adapt an uncapacitated VRPTW formulation and use a combination of robust optimization in a first phase master problem and stochastic programming with recourse for daily schedules to address the uncertainty in service times and customer occurrence. Given historical demand data, we consider each day in the planning horizon as a scenario and generate a robust master plan for customers who are most likely (ML) to occur. The objective of the recourse actions is, for each scenario, to maximize the coverage of customers, to minimize the total time spent by the couriers and the total earliness and lateness penalty, and to maximize the similarity of the daily routes with the robust master plan.

In formulating a master plan, a variety of approaches could be used in addressing uncertainty in service times and customers. However, because one of the primary goals in generating the master plan is to improve similarities in daily schedules, which vary with the uncertainty outcome, an intuitive approach is to use a method that would result in a master plan that would stay *good* for all possible realizations of the uncertainty, and thus would require few modifications to adapt to the daily schedules. This suggests using a robust optimization approach to plan the master schedule, as it would exhibit little sensitivity to data variations. Likewise, considering the customers ML to occur when constructing the master plan should improve the similarity of daily schedules, because the master plan would be tailored toward these customers who appear in most days.

To solve the CDP, we develop a two-phase approximate solution procedure, called master and daily scheduler (MADS), to obtain solutions for large-scale real-life problems. This insertion-based heuristic generates a master plan, which is then used to generate daily schedules using the recourse actions and improved with a tabu search. Tabu search has been used with good results on vehicle routing problem (VRP) (Cordeau and Laporte 2004).

1.2. Organization of the Paper

In the remainder of this introductory section, we present a related literature review. In §2, we present the CDP formulation for problems with service time uncertainty and probabilistic customers. In §3, we propose the two-phase heuristic for MADS and present the tabu search improvement procedure. We present our computational results in §4. These include a comparison of MADS to solutions obtained by independently scheduling each day without a master plan, to current industry practice that executes territory planning, and to ConVRP on benchmark problems. We finish the paper with a summary and conclusions in §5.

1.3. Literature Review

The VRP variants related to our work are VRP with stochastic demands (VRPSD), with stochastic customers (VRPSC), and with stochastic service and travel times (VRPSSTT). A major contribution to VRPSD comes from Bertsimas (1992), where a priori solutions use different recourse policies to solve the VRPSD and bounds, and asymptotic results and other theoretical properties are derived.

A number of models and solution procedures for VRPSD and VRPSC allow recourse actions to adjust an a priori solution after the uncertainty is revealed. Different recourse actions have been proposed in the literature, such as skipping nonoccurring customers, returning to the depot when the capacity is exceeded, or complete reschedule for occurring customers (Jaillet 1988; Bertsimas, Jaillet, and Odoni 1990; Waters 1989). Recent work by Morales (2006) uses robust optimization for the VRPSD with recourse. It considers that vehicles replenish at the depot, computes the worst-case value for the recourse action by finding the longest path on an augmented network, and solves the problem with a tabu search heuristic. Sampling methods are also popular in solving stochastic VRP (Birge and Louveaux 1997). Recently, Hvattum, Løkketangen, and Laporte (2006) develop a heuristic method to solve a dynamic and stochastic VRP problem, where sample scenarios are generated, solved heuristically, and combined to form an overall solution. Compared with stochastic customers and demands, the VRPSSTT has received less attention. Laporte, Louveaux, and Mercure (1992) propose three models for VRPSSTT: chance constrained model, three-index recourse model and two-index recourse model, and present a general branch-and-cut algorithm for all three models. The VRPSSTT model is applied to a banking problem and solved with an adaptation of the savings algorithm (Lambert, Laporte, and Louveaux 1993). Jula, Dessouky, and Ioannou (2006) develop a procedure to estimate the arrival time to the nodes in the presence of hard time windows. These estimates are embedded in a dynamic programming algorithm to determine the optimal routes.

Recent work on the CDP has modeled customer service for fixed route delivery systems under stochastic demand (Haughton and Stenger 1998). Later, Haughton (2000) develops a framework for quantifying the benefits of route re-optimization, again under stochastic customer demands. Zhong, Hall, and Dessouky (2007) propose an efficient way of designing driver service territories considering uncertain customer locations and demand. Their method uses a two-stage model to construct core service territories in the strategic level and assigns customers in the noncore territories on a daily basis to adapt to the uncertainty in the operational level. The territory model is based on approximation equations of the distance traveled. The operational level makes it possible for all drivers to share their capacities by introducing the concept of "flex zone." This approach, however, does not consider customer time windows. Groër, Golden, and Wasil (2008) introduce the ConVRP model. The objective is to obtain routes such that the same drivers visit the same customers at roughly the same time on each day that the customers need service. They develop an algorithm, ConRTR (ConVRP Record-to-Record travel), which first generates a template and from it generates daily schedules by skipping nonoccurring customers and inserting new customers.

Robust optimization methodology was introduced by Ben-Tal and Nemirovski (1998, 1999) and El-Ghaoui, Oustry, and Lebret (1998) for convex programs, which is recently extended to integer programming by Bertsimas and Sim (2003). The general approach of robust optimization is to optimize against the worst instance because of data uncertainty by using a min-max objective. This typically results in solutions that exhibit little sensitivity to data variations and are said to be *immunized* to this uncertainty. Thus, robust solutions are *good* for all possible data uncertainty. Robust solutions are likely to be efficient, because they tend not to be far from the optimal solution of the deterministic problem and significantly outperform the deterministic optimal solution in the worst case (Goldfarb and Iyengar 2003; Bertsimas and Sim 2004).

The robust optimization methodology assumes the uncertain parameters belong to a given bounded uncertainty set. For fairly general uncertainty sets and optimization problems, the resulting robust counterpart can have a comparable complexity to the original problem. This nice complexity result, however, does not carry over to robust models of problems with recourse, where linear programs with polyhedral uncertainty can result in NP-hard problems (Ben-Tal et al. 2004). An important question, therefore, is how to formulate a robust problem that is not more difficult to solve than its deterministic counterpart. In particular, Sungur (2009) and Sungur, Ordóñez, and Dessouky (2008) show that obtaining robust solutions for VRP with demand and travel time uncertainty is not more difficult than obtaining the deterministic solutions.

2. CDP Formulation

In this section, we formulate the CDP as a MIP model. The delivery requests arrive daily from potential customers with known time windows but uncertain service times at the beginning of each day. The locations of the customers are known, but it is not known a priori whether a particular customer requests a delivery at a given day. There are a limited number of couriers to route for a limited time period each day. The first goal is to construct an a priori master plan for the planning horizon to be used in constructing daily schedules by adapting to the daily customer requests. The second goal is to modify the master plan to construct daily schedules for couriers to serve as many customers as possible, while maintaining route similarity and at the same time minimizing earliness/lateness penalties and the total time spent by the vehicles in each day, which accounts for travel, waiting, and service times.

We measure the similarity of a route on the daily schedule and a route on the master plan by counting the number of customers of the daily route that are within a given distance of any customer on that master plan route. The similarity of the daily schedule is given by assigning each daily schedule route to a master plan route, to represent the same driver, so that the overall similarity measure is maximized. The larger the measure, the more nodes are within a given distance of the corresponding master route, and accordingly, the more nodes are visited by the same driver. This measure captures some important aspects of territory familiarity and the visiting frequency to a customer by the same driver (Zhong, Hall, and Dessouky 2007). They describe a driver learning model, which shows that when the number of visits to a particular cell by the same driver increases, the average time spent to serve each stop in this cell approaches a lower limit. In addition, when a customer is visited by the same driver, the service quality also improves.

Given a total of D days of historical data, we consider each past day (scenario) as a realization of uncertainty, and we construct scenario-based uncertainty sets for service times and customer occurrence. In constructing the master plan, we address the probabilistic nature of the customers by attempting to serve only the customers with high frequency of occurrence and use a robust optimization approach to represent the uncertain service times; both ideas aim to improve the similarity in daily schedules. Hence we obtain robust a priori routes for the master plan, which are then used as the starting point to construct the daily routes to serve the observed demand optimizing the combined time, penalty and similarity objective. The goal is that the robust master plan that has been *trained* by the past data or scenarios can then be used to generate future daily schedules. The implicit assumption is that the scenarios considered are representative of future demand.

We formulate the CDP based on the uncapacitated VRP with soft time windows. Similar to the classic VRPTW, given a network of customers and the location of the depot, the objective is to route the fleet of vehicles to serve the maximum number of customers based on their service times and time windows. We allow a vehicle to arrive before or after the time window at a given penalty and consider the uncapacitated problem because packages are small relative to vehicle capacity. The lack of capacity constraint implies that every customer will be served by only one vehicle. The vehicles start and end their routes at the depot. The length of a route, or total time spent, is composed of travel, waiting, and service times. There is a common due date for the vehicles to return to the depot, which is a hard constraint. We consider one routing problem for the master plan that serves high frequency customers and worst-case (WC) service times, and one routing problem for each of D scenarios of daily schedules. These $D+1$ routing problems are related by keeping track of the similarity of daily routes to the master plan. Thus the size of the CDP is $D+1$ times the size of a VRPTW.

We now introduce the mathematical formulation of the CDP, which is given below in problem 1. We begin by setting the notation. The depot is located at node 0. Let K be the set of couriers, and N_D the set of integers from 0 to D to indicate scenarios, including master plan as scenario 0. There are a total of n customers indexed by $C = \{1, 2, \ldots, n\}$, a depot node 0, and $|K|$ artificial nodes indexed by $A = \{n+1, \ldots, n+|K|\}$. The frequency of occurrence of customer i, p_i, is defined as the ratio of the number of days customer i appears over the total number of days D, which is also referred to as the probability of occurrence of this customer. Let C^d be the set of customers who occur in a given scenario d, and V^d the set of all the nodes that occur in a given scenario d, $V^d = C^d \cup A \cup \{0\}$. The time to traverse the arc from node i to node j is given by t_{ij} and s_i^d is the service time of node i in scenario d. In particular, $\forall\, d \in N_D$ the following is true: $s_0^d = 0$; $\forall\, i \in A$, $k \in K$, $t_{i0k} = t_{0ik} = s_i^d = 0$; $\forall\, i \in A$, $\forall\, j \in C^d$, $t_{ij} = t_{0j}$, and $t_{ji} = t_{j0}$; and $\forall\, i, j \in A$, $i \neq j$, $t_{ij} = t_{ji} = 0$. We let R be the time threshold to consider two nodes *near* each other, and keep track of which nodes are near with the parameter $v_{ij} = 1$ if $t_{ij} \leq R$; otherwise 0. The value of a_i^d represents the earliest start time and b_i^d is the latest start time to serve customer i in scenario d. The common due date for all vehicles is L, which is also referred as the route length. Let M be a sufficiently large number.

If the arc from node i to node j is traversed by vehicle k in scenario d, then the binary variable $x_{ijk}^d = 1$; otherwise 0. If customer i is visited by vehicle k in scenario d, then binary variable $z_{ik}^d = 1$; otherwise 0. The continuous variable y_{ik}^d is the arrival time to node i in scenario d by vehicle k except the depot in which case it is the departure time, i.e., $y_{0k}^d = 0$. In particular, y_{ik}^d for $i \in A$ corresponds to the arrival time to the depot of vehicle k. Note that $y_{ik}^d = 0$ for a customer i, which is not served by vehicle k, i.e., when $z_{ik}^d = 0$. The continuous variable e_{ik}^d is the earliness penalty and l_{ik}^d is the lateness penalty of customer i for vehicle k in scenario d; similarly $e_{ik}^d = l_{ik}^d = 0$ when $z_{ik}^d = 0$.

To measure the similarity between the daily schedule of scenario d and the master plan, we assign each daily schedule route to a master route (same driver) and count how many of the nodes in each daily schedule route are within R of some node on their assigned master route. We use the assignment that maximizes the overall similarity. For this, we need to use two auxiliary sets of binary variables, m_{kl}^d and r_{il}. If route k of scenario d is assigned to the master plan route l, then variable $m_{kl}^d = 1$; otherwise 0. If customer i is near any node of master route l, then $r_{il} = 1$; otherwise 0. In other words, $r_{il} = 1$ if and only if $v_{ij} = 1$ for at least one node j in route l. We say node i is *good* if the vehicle k serving it (i.e., $z_{ik}^d = 1$) is assigned to a master route l (i.e., $m_{kl}^d = 1$), for which the node is near (i.e., $r_{il} = 1$). That is, the binary variable $g_{ikl}^d = z_{ik}^d m_{kl}^d r_{il} = 1$ if i is good; otherwise 0. Only when all of m_{kl}^d, z_{ik}^d, and r_{il} are 1, g_{ikl}^d is 1. We linearize the expression of g_{ikl}^d in the model. The similarity measure is the total number of nodes that are good. Problem 1 is presented below.

The CDP objective function is

$$\min \sum_{d \in N_D} \sum_{k \in K} \left(-\alpha_1 \sum_{i \in C^d} z_{ik}^d + \alpha_2 \sum_{i \in A} y_{ik}^d + \alpha_3 \sum_{i \in C^d} l_{ik}^d + \alpha_4 \sum_{i \in C^d} e_{ik}^d \right) - \sum_{d \in N_D \setminus \{0\}} \alpha_5 \sum_{i \in V^d} \sum_{k \in K} \sum_{l \in K} g_{ikl}^d. \tag{1}$$

Routing constraints are

$$\text{s.t.} \quad \sum_{j\in V^d,\, i\neq j} x^d_{jik} = z^d_{ik} \quad i\in C^d,\ k\in K,\ d\in N_D, \tag{2}$$

$$\sum_{j\in V^d,\, j\neq i} x^d_{ijk} = z^d_{ik} \quad i\in C^d,\ k\in K,\ d\in N_D, \tag{3}$$

$$\sum_{i\in V^d\setminus\{0\}} x^d_{0ik} = 1 \quad k\in K,\ d\in N_D, \tag{4}$$

$$\sum_{i\in V^d\setminus\{0\}} x^d_{i0k} = 1 \quad k\in K,\ d\in N_D, \tag{5}$$

$$x^d_{i0k} = 1 \quad i\in A,\ k\in K,\ d\in N_D. \tag{6}$$

Time and time window violation definitions are

$$\text{s.t.} \quad y^d_{ik} + t_{ij} + s^d_i \le y^d_{jk} + M(1 - x^d_{ijk})$$
$$i\in V^d,\ j\in V^d\setminus\{0\},\ i\neq j,\ k\in K,\ d\in N_D, \tag{7}$$

$$a^d_{ik} \le y^d_{ik} + e^d_{ik} + M(1 - z^d_{ik}) \quad i\in C^d,\, k\in K,\, d\in N_D, \tag{8}$$

$$y^d_{ik} \le b^d_{ik} + l^d_{ik} \quad i\in C^d,\ k\in K,\ d\in N_D, \tag{9}$$

$$y^d_{ik} \le L z^d_{ik} \quad i\in C^d,\ d\in N_D. \tag{10}$$

Similarity measure constraints are

$$\text{s.t.} \quad \sum_{j\in V^d} v_{ij} z^0_{jl} \ge r_{il} \quad i\in C^d,\ l\in K, \tag{11}$$

$$\sum_{j\in V^d} v_{ij} z^0_{jl} \le M r_{il} \quad i\in C^d,\ l\in K, \tag{12}$$

$$g^d_{ikl} \ge z^d_{ik} + m^d_{kl} + r_{il} - 2$$
$$i\in V^d,\ k\in K,\ l\in K,\ d\in N_D\setminus\{0\}, \tag{13}$$

$$g^d_{ikl} \le (z^d_{ik} + m^d_{kl} + r_{il})/3$$
$$i\in V^d,\ k\in K,\ l\in K,\ d\in N_D\setminus\{0\}, \tag{14}$$

$$\sum_{k\in K} m^d_{kl} = 1 \quad l\in K,\ d\in N_D\setminus\{0\}, \tag{15}$$

$$\sum_{l\in K} m^d_{kl} = 1 \quad k\in K,\ d\in N_D\setminus\{0\}. \tag{16}$$

Domain constraints are

$$\text{s.t.} \quad e^d_{ik} \ge 0 \quad i\in C^d,\ k\in K,\ d\in N_D, \tag{17}$$

$$l^d_{ik} \ge 0 \quad i\in C^d,\ k\in K,\ d\in N_D, \tag{18}$$

$$x^d_{ijk} \in \{0,1\} \quad i,j\in V^d,\ i\neq j,\ k\in K,\ d\in N_D, \tag{19}$$

$$y^d_{ik} \ge 0 \quad i\in V^d,\ k\in K,\ d\in N_D, \tag{20}$$

$$z^d_{ik} \in \{0,1\} \quad i\in C^d,\ k\in K,\ d\in N_D, \tag{21}$$

$$r_{il} \in \{0,1\} \quad i\in V^d,\ l\in K, \tag{22}$$

$$m^d_{kl} \in \{0,1\} \quad k\in K,\ l\in K,\ d\in N_D\setminus\{0\}, \tag{23}$$

$$g^d_{ikl} \in \{0,1\} \quad i\in V^d,\, k\in K,\, l\in K,\, d\in N_D\setminus\{0\}. \tag{24}$$

The objective function (1) maximizes the number of customers served and minimizes the total time spent by the vehicles as well as the total earliness and lateness penalty, for each scenario, including the master plan. In addition, the weighted similarity of the scenario routes with the master plan is also maximized. We consider a set of positive weights $\alpha_1, \ldots, \alpha_5$ to balance these competing objectives. Although the specific values of the weights depend on the problem and the planner's objectives, it is reasonable to consider a higher value of α_1, so that not visiting a customer to avoid travel time or earliness or lateness penalties would not be desirable. Note that in many real-life cases, $\alpha_4 = 0$ because there is no explicit penalty for waiting time, but it indirectly increases the total time spent.

Constraints (2)–(5) are the routing constraints. Constraint (6) forces every artificial node to be visited at the end of the route to keep track of the time spent by the vehicles. Constraint (7) defines y^d_{jk}, the arrival time at j when customer j is served right after customer i for vehicle k in scenario d. The increasing time would also guarantee that there are no subtours in the solution. Constraints (8)–(9) impose the earliness and lateness penalty. Constraint (10) imposes the common due date of the vehicles. Constraints (11)–(12) ensure that only when the distance between node i and at least one of the nodes in master route l is less than R, $r_{il} = 1$; otherwise 0. Constraints (13)–(14) is the linearization of $g^d_{ikl} = z^d_{ik} m^d_{kl} r_{il}$. Constraints (15)–(16) ensure every daily route is assigned to a different master route. Last, constraints (17)–(24) are bounds on the variables. Note that constraints (11)–(16) are the linking constraints, relating each scenario d with the master plan. Removing these constraints separates the CDP problem into $D+1$ unrelated VRPTW.

For the master plan (scenario 0) in the CDP, we need to define the set of customers C^0 and the value of uncertain service time s^0_i for each customer i in this set. For the former, we select the customers with highest probability of occurrence p_i. For the latter, we use robust optimization to construct WC service time values for the master plan. For customer i, we assume that the possible realizations of service times are in the convex hull of the scenario realizations $\{s^d_i\}_{d\in N_D}$. With this model of uncertainty, the WC service time considered by robust optimization is simply $s^0_i = \max_{d\in N_D} s^d_i$. A general treatment of robust optimization for VRP with scenario-based uncertainty appears in Sungur, Ordóñez, and Dessouky (2008).

The formulation is different from the conventional VRP in that (1) to get a robust master plan, the service time we are using is the worse-case service time; (2) the model has multiple objectives, because it considers not only the number of customers served, total time spent, earliness penalty and lateness penalty, but also the similarity of the daily routes with the master routes; and (3) the solution of problem 1 includes both daily routes and a master plan.

3. MADS Heuristic

Because VRP is NP-hard, it is clear that the formulation of CDP (problem 1) is NP-hard since the VRP is a special case when $\alpha_1 = \alpha_3 = \alpha_4 = \alpha_5 = 0$. To address the challenge of solving a large-scale real-life CDP, we develop a heuristic solution procedure, MADS, based on insertion and tabu search. The approach is to separate the CDP into $D+1$ problems by removing constraints (11)–(16) and using an insertion heuristic to solve each of these problems. We coordinate the solutions of the master plan and scenarios in a two-phase iterative process. The master plan is used to construct routes for each scenario with a partial rescheduling recourse. This partial rescheduling recourse combines skipping customers not present in the scenario and inserting the new customers. The MADS output is a robust master plan and daily schedules for the given set of scenarios. This master plan can then be used to generate daily schedules, once the uncertainty of a given day is realized, through the partial rescheduling recourse; the daily schedule is then improved with a tabu search heuristic.

When inserting customers to routes, we use an insertion routine (Algorithm 1) that greedily minimizes the cost of insertion. That is, the cheapest among all feasible insertions is done at each step. When scheduling the master plan, the cost of insertion of customer j in the route of vehicle k is determined by the increase in the time spent and the increase in the total penalty of all the customers served by that vehicle, i.e., $\Delta(\alpha_2 \sum_{i\in A} y_{ik}^0 + \alpha_3 \sum_{i\in H^0} e_{ik}^0 + \alpha_4 \sum_{i\in H^0} l_{ik}^0)$, where H^d is the set of scheduled customers for $d \in N_D$. When scheduling daily routes, however, the cost of insertion is determined by not only the increase in the time spent and the total penalty, but also the decrease of similarity, i.e., $\Delta(\alpha_2 \sum_{i\in A} y_{ik}^d + \alpha_3 \sum_{i\in H^d} e_{ik}^d + \alpha_4 \sum_{i\in H^d} l_{ik}^d - \alpha_5 \sum_{i\in H^d} r_{ik})$. Without loss of generality, we are constructing daily route k according to master route k.

Algorithm 1 (Insertion routine)

Require: Initial routes, set of unscheduled customers
Calculate insertion cost of possible insertion locations for nonscheduled customers
repeat
Pick the cheapest insertion
Update routes
Update insertion cost
until All customers inserted or no feasible insertion possible
return The resulting routes

The first phase (Algorithm 2) is the construction of an initial solution. An initial robust master plan is constructed by starting with empty routes and making insertions of customers using Algorithm 1. Then, the routes of the master plan are updated by each scenario following the recourse actions to construct daily schedules. Each scenario starts to adapt the robust master routes by omitting (not visiting) the customers who do not occur in that particular scenario and visiting the remaining customers following the same sequence as in the master routes. Then greedy insertion, Algorithm 1, is used to insert the new customers of that particular day that do not occur in the master routes.

Algorithm 2 (Phase One)

Require: Distance matrix, master data, scenario data, maximum route length
Call insertion routine for master plan starting with empty routes
for Each scenario d **do**
Drop nonoccurring customers from master routes
Call insertion routine for new customers
end for
Calculate the objective value and save the current solution
return The current solution

The second phase (Algorithm 3) is iterative. At each iteration, first, the scenarios give feedback to the master plan about the customers who could not be scheduled in their daily routes; second, based on this feedback, the master plan prioritizes these unscheduled customers by the number of scenarios that they appear but could not be scheduled. Then, the master plan updates its routes by performing feasible maximum priority insertions in the cheapest way. Note that the selection of customers is based on the priority not on the cost of insertion. However, once a customer is selected, the cheapest possible insertion is done for this particular customer. Then, the new master plan is re-dispatched to the scenarios, which construct their daily schedules with respect to the recourse action as before. At the end of each iteration, the objective function is evaluated as

$$\sum_{d\in N^d\setminus\{0\}} \sum_{k\in K} \left(-\alpha_1 \sum_{i\in H^d} z_{ik}^d + \alpha_2 \sum_{i\in A} y_{ik}^d + \alpha_3 \sum_{i\in H^d} e_{ik}^d + \alpha_4 \sum_{i\in H^d} l_{ik}^d \right) - \sum_{d\in N_d\setminus\{0\}} \alpha_5 S_d,$$

where S_d (the similarity of scenario d) is obtained by solving a maximum assignment problem. The problem is to assign the daily routes to the master routes optimally to get maximum similarity. The iterations of the second phase, and thus the overall two-phase algorithm stop when there is no improvement in the objective.

Algorithm 3 (Phase Two)
 Require: Solution of Phase One, maximum route length
  repeat
   Calculate priorities of customers not served in the scenarios
   repeat
    In decreasing priority order, pick a customer and call insertion routine to insert it in master route
   until No priority customer or no feasible insertion possible
   for Each scenario d, **do**
    Take the master routes as initial solution
    Drop the nonoccurring customers
    Call insertion routine for new customers
   end for
   Calculate the objective value
   if The objective value is improved, **then**
    Save the current solution
   end if
  until The objective value is not improved
  return The current solution

Note that, for the master plan, the first phase is cost driven, whereas the second phase is priority driven. For the scenarios, both phases are cost driven based on the current master plan. The maximization of the number of customers served is mainly because of the recourse action of partial rescheduling of the routes; the minimization of the time spent and total penalty is mainly because of the greedy insertions; and maximization of the similarity of routes is because of generating daily schedules based on a common master plan and the feedback procedure between the master plan and daily schedules to prioritize the customers in the iterative second phase.

The structure of the algorithm prioritizes serving all customers, and then focuses on similarity, time spent, and penalties when making insertions in the construction of routes. The algorithm makes all feasible insertions possible regardless of the impact on cost. This behavior corresponds to considering the weight α_1 much larger than $\alpha_2, \ldots, \alpha_5$ in the CDP (problem 1). The setting of the remaining weights is problem dependent and not structural to the algorithm. Although the algorithm builds the daily schedules by modifying the master route, the degree to which this favors similarity is given by the weights used in the insertion.

We also consider a *buffer capacity* between Phases One and Two of the algorithm. In the first phase, we reserve a buffer capacity by decreasing the common due date of the vehicles. This slack time is later used in the second stage to schedule additional customers. This parameter of the algorithm is actually a tool to balance the cost-driven stage and the priority-driven stage in constructing the master plan, which indirectly effects the daily schedules as well. The algorithm must also identify a subset of the customers to be considered in the master problem (C^0) during the first phase, because in a real-life instance, the total number of customers is too large to be feasibly scheduled. In the computational section, we explore experimentally the effect of these algorithmic decisions on the quality of the solution.

The description of MADS is given by Algorithms 1–3. Note that the output is a master plan that depends on the given set of scenarios. This master plan is used to create daily schedules using the partial rescheduling recourse: dropping nonoccurring customers and then using Algorithm 1 to insert the remaining customers. We then use a tabu search algorithm (Algorithm 4) to improve the daily routes obtained by the MADS heuristic. The tabu search algorithm is not applied to the master plan. This implementation of the tabu search considers the neighborhoods obtained from the standard two-opt exchange move and the λ-interchange move. The algorithm evaluates solutions based on the objective function of the CDP (problem 1), i.e., the weighted sum of the number of customers served, time spent, earliness penalty, lateness penalty, and similarity.

Algorithm 4 (Tabu search algorithm)
 Require: Solution of Phase Two
  for Each scenario d, **do**
   repeat
    Randomly choose two routes from the solution
    Generate $\eta_{\max}$ neighbors from λ-interchange operator
    Generate $\gamma_{\max}$ neighbors from two-opt operator
    Choose the best solution and make the move
    Randomly generate tabu tenure θ from a uniform distribution $U(\theta_{\min}, \theta_{\max})$;
    if The move is λ-interchange, **then**
     Make moving the exchanged nodes tabu for θ iterations
    else
     Make removing the new arcs tabu for θ iterations
    end if
   until No improvement in $I_{\max}$ iterations
  end for
  Calculate the objective value and save the current solution
  return The current solution

At each iteration, the tabu search generates $\eta_{\max}$ λ-interchange neighbors and $\gamma_{\max}$ two-opt neighbors of the current solution. These neighborhoods are created forbidding certain moves, referred to as tabu, for a given number of iterations θ. In our implementation, the number of tabu iterations is randomly (RD) generated uniformly in $(\theta_{\min}, \theta_{\max})$. For λ-interchange move, feasible moves from a solution consider that up to P and Q nodes are exchanged between two routes of the solution. The tabu search at each iteration moves to the best neighbor, temporarily allowing a move to a worse solution to escape the local optima. The tabu status is overridden if the new solution is better than the best solution so far and the algorithm terminates if there is no improvement in $I_{\max}$ iterations.

4. Experimental Analysis

We present two sets of experiments. In the first set of experiments, we analyze the sensitivity of the MADS algorithm to the problem data and algorithm settings. Specifically, we present results that show how the algorithm varies with changes in the number of customers of the master problem, the type of master problem considered, the sample size for training the master plan, the service time distribution, the probability distribution of customer occurrence, and the weight of similarity α_5. We compare the quality of the solution of the MADS algorithm with a solution obtained by independently scheduling scenarios without a master plan, which we refer as the independent daily insertion (IDI) algorithm.

The IDI algorithm treats each scenario as an independent CDP and there is no master plan created. The insertion routine, Algorithm 1, is executed for each day with initially empty routes, without reserving a buffer capacity, and with the cost of insertion given by the increase in time spent and penalties. Then the tabu search, Algorithm 4, is executed to improve the solution obtained from Algorithm 1. Therefore the IDI algorithm does not make any considerations in increasing the similarity of the resulting independent daily schedules, but maximizes the number of customers served while minimizing the sum of time spent and penalties for each day.

In the second set of experiments, we evaluate the performance of MADS in obtaining solutions to two large-scale real-world CDP instances from UPS. We compare the solution of the MADS algorithm with the current practice of a courier delivery company. In addition, we analyze the sensitivity of the algorithm to different settings of the buffer capacity. We also compare the results of a modified version of MADS with ConRTR over a set of ConVRP benchmark problems.

Regarding the objective function weights of the CDP problem considered in the formulation, we follow two guidelines: (1) we consider that satisfying all customers is an overriding objective, therefore we set α_1 much higher than $\alpha_2, \ldots, \alpha_5$ and (2) we consider similarity more important than time spent and penalties. The idea is to study the trade-off between the similarity and other operational costs, and the IDI solution provides a benchmark with a solution that does not include similarity. In the computational results, we set the objective function weights as $\alpha_1 = 10{,}000$, $\alpha_5 = 5$, $\alpha_2 = \alpha_3 = 1$, and $\alpha_4 = 0$ because our application does not consider an explicit earliness penalty.

Throughout the experimental analysis, we separate the scenarios available in two groups. We use the data for the first group to *train* the master plan and we use the remaining scenarios to *evaluate* the performance by treating it as future outcomes. In addition, the values of the parameters for the tabu search algorithm (Algorithm 4) are as follows: $I_{\max} = 200$; $\eta_{\max} = \gamma_{\max} = 100$; $P = Q = 2$; $\theta_{\min} = 10$; and $\theta_{\max} = 20$. The threshold for computing the route similarity is set as $R = 0.1$ mile. Finally, all experiments are performed on a Dell Precision 670 computer with a 3.2 GHz Intel Xeon processor and 2 GB RAM running Red Hat Linux 9.0, and all the solutions could be obtained within one hour of CPU time.

4.1. Problem Data and Uncertainty

The CDP data obtained concerns operations of a large courier company (UPS) in an urban area with known customer locations. At the beginning of each day, any of these potential customers can put a delivery request with an uncertain service time. The travel time is considered deterministic and to convert the distance measures to time units, it is assumed that the couriers travel at an average speed of 35 mph in the city. We have two data sets for this application, which are described in Table 1.

We now analyze the service time characteristics and customer frequency distribution for data set 1. Similar trends are observed for data set 2. As it is common in the routing literature and in industry, service times follow a lognormal distribution, see Dessouky et al. (1999). In our particular case, we observe that

Table 1 Description of the Two Data Sets

Variables	Data set 1	Data set 2
Number of potential customers	3,715	5,178
Average number of customers/day	472	610
Total number of days	29	42
Planning horizon	14 or 15 days	21 days
Number of couriers	4	5
Operation time of couriers	9:00 A.M.–8:00 P.M.	9:00 A.M.–9:00 P.M.
Total number of service requests	13,688	25,631

the service times of the customers in data set 1 are closely approximated with a shifted and scaled lognormal distribution with mean 0.0953 and standard deviation 0.25. The mean and the standard deviation of the actual service times are, respectively, about six and four minutes. The frequency of occurrence distribution for the customers of data set 1 is depicted in line P2 of Figure 1. The line P4 is its continuous approximation, which is a shifted power function. Note that there are customers with probability 1 (i.e., occurring in all of the scenarios). In addition, we present two more continuous distributions, P1 and P3, which are generated by modifying P4. In P1, the probabilities of occurrence are decreased with respect to the original P2; and in P3, they are increased. These three distributions, P1, P2, and P3, are used in our experiments to sample scenarios.

4.2. Sensitivity of MADS

In this first set of experiments, we only focus on data set 1 for space considerations. We explore the effect of changing algorithmic parameters and problem data on the MADS algorithm and we compare the quality of the solutions with the IDI algorithm. For the current experiments, we fix the value of the buffer capacity and the set of customers to be scheduled in the master plan during Phase One of the MADS algorithm. We set the buffer capacity to 50%, meaning that only half of the total allowed time for the couriers is considered during the first phase. In this way, we weigh the cost-driven first phase and priority-driven second phase equally. For the set of master plan customers, we refer to the distribution P2 in Figure 1 and define a cut to select the customers with high probability of occurrence. Customers with a probability higher than the cut value are selected.

For the uncertainty in service time and probabilistic customers, we consider a base case with respect to our fitted lognormal distribution and the probability distribution P2 in Figure 1. For each problem instance, we sample data of scenarios for the planning horizons with respect to these two distributions. First, the occurrence data of each scenario is generated by RD selecting customers according to P2 until 472 customers are selected in each scenario. Then, each customer is assigned a random service time following the lognormal distribution. Recall that the time windows and travel times are deterministic. Thus all the required data for a problem instance is generated by this process. Also, recall that for the MADS algorithm, only the first half of the total data (15 days) is used to generate the master plan for a planning horizon and the remaining half (14 days) is used to evaluate its performance; whereas for the IDI algorithm, only the second half is used as *future* outcomes.

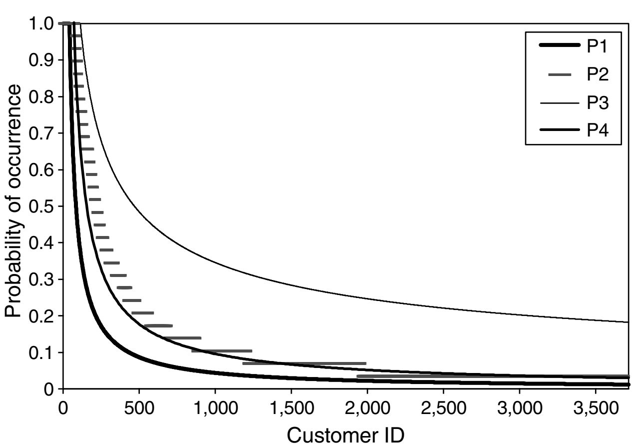

Figure 1 **Actual and Modified Probability Distributions of the Occurrence of Customers**

A set of experiments is done to choose a cut value for the MADS algorithm and the results are shown in Table 2. For each case, we generate 30 random problem instances and report the average of the solutions. The instances generate customers following the P2 distribution in Figure 1 and with service times following the fitted lognormal distribution with standard deviation $\sigma_s = 0.250$. The column "Cut" in Table 2 indicates the cut value to determine the set of customers to be initially scheduled in master plan. The remaining five columns report the average solution results: "NS" is the total number of customers who could not be served in the daily schedules; "Time" is the total time spent by the couriers in the daily schedules (composed of travel, waiting, and service times); "Penalty" is the total lateness penalty in the daily schedules; "Sim" is the node similarity measure (i.e., the total number of nodes that are good); and "Obj" is the value of the CDP objective function. From the table, we can see that the solution is best when the cut is 0.5. When the cut value changes, the weighted change in similarity is larger than the change in time spent and penalties, therefore similarity plays a much larger role in the objective value. When the cut is too low, the master plan schedules a lot of low probability customers, resulting in low similarity, therefore the objective value is high. However, when the cut is too high, most customers are inserted in Phase Two,

Table 2 **MADS Sensitivity to the Cut Value**

Cut	NS	Time	Penalty	Sim	Obj
0.10	0	95,884	6	6,034	65,699
0.30	0	97,037	1	6,328	65,402
0.50	0	97,108	0	6,426	64,987
0.70	0	97,027	0	6,328	65,400
0.90	0	97,125	0	6,188	66,214

Note. Customers follow distribution P2; standard deviation of service times $\sigma_s = 0.250$.

which is priority driven, resulting in fewer total customers in the master plan and accordingly low similarity, so the objective value increases again. Therefore we use a cut of 0.5 in later experiments.

When the tabu search algorithm is applied in this first set of experiments, we observe that the time spent of the initial solution is reduced by about 5%. The tabu search also reduces the penalty by about 99%, making the penalty almost negligible. The reason is that it changes the positions of the customers with high penalty in the initial solution through λ-interchange moves or two-opt moves, so that almost all time windows are satisfied. Last, we observe that the tabu search also improves the similarity. Overall, we can see that the tabu search is effective in improving the initial solution. Similar effects are observed in later experiments.

Another set of experiments is done to explore the effect of using different methods to choose initial customers for the master plan, and calculate service time of these customers. We compare three methods of choosing initial customers: choosing ML customers, choosing customers RD, and choosing customers with the longest service time (LS). In terms of service time, we compare two methods: using WC service time and average service time (AVG). This leads to six possible combinations of how to define the master plan. We refer to these combinations with the acronyms above, for example, WC-ML means using WC service time and choosing ML customers.

The results are shown in Table 3. The column "SD" gives the standard deviation of the objective value over the 30 instances. From the results, we conclude that solutions with WC service times have less variance than those with AVG service times for the same method of choosing initial customers. The reason is that considering WC service times in the master plan builds routes with more slack that are better suited to adjust to different scenarios, giving less variance. In addition, the objective value for WC-ML is better than that for AVG-ML, and the ML method yields better results than both RD and LS. Therefore, in later experiments, we use WC-ML.

Table 3 **MADS Sensitivity to Different Methods for Master Plan**

Method	NS	Time	Penalty	Sim	Obj	SD
WC-ML	0	97,108	0	6,426	64,987	562
WC-RD	0	97,592	1,008	6,132	67,916	808
WC-LS	0	96,510	2	6,104	65,967	517
AVG-ML	0	98,146	67	5,726	69,881	1,604
AVG-RD	0	98,209	911	5,334	70,067	964
AVG-LS	0	96,308	200	6,216	65,404	721

Note. Customers follow distribution P2; standard deviation of service times $\sigma_s = 0.250$.

Table 4 **MADS Sensitivity to the Sample Size for Training the Master Plan**

Sample	NS	Time	Penalty	Sim	Obj
7–22	0	97,117	1	6,412	65,039
15–14	0	97,108	0	6,426	64,987
22–7	0	97,124	0	6,384	65,205

Note. Customers follow distribution P2; standard deviation of service times $\sigma_s = 0.250$.

A set of experiments is done to explore the effect of the sample size to train the master plan. We compare three variants: 7–22, 15–14, and 22–7, where the first number is the number of days used to train the master plan, and the second number is the number of days used to evaluate the master plan. The results are shown in Table 4. From the table, we can see that 15–14 produces the best result. It means that a sample size of 15 days is enough to produce a good solution. Thus we use 15–14 in later experiments.

When comparing MADS with IDI, we generate additional cases by deviating from the base case in two ways. First, we change the standard deviation σ_s of the lognormal distribution to see the effect of increased service times, with $\sigma_s = 0.500$, and decreased service times, with $\sigma_s = 0.125$. Second, instead of P2, we sample customers from P1 and P3 in Figure 1. When moving from one case to another, we modify only one parameter at a time keeping the rest of the problem instance the same, which allows observing the sole effect of changing this particular parameter. Table 5 provides these experimental results. The left part is the input parameters and the right part is the output measures. Because the IDI algorithm does not generate a master plan, we calculate the similarity of its solution based on the master

Table 5 **Comparison of MADS with IDI**

Alg	Prob	Std	NS	Time	Penalty	Sim	Obj
MADS	P1	0.125	0	94,391	0	6,314	62,854
MADS	P1	0.250	0	97,067	0	6,412	64,979
MADS	P1	0.500	12	107,741	18	5,544	80,035
MADS	P2	0.125	0	94,874	39	6,146	64,168
MADS	P2	0.250	0	97,108	0	6,426	64,987
MADS	P2	0.500	12	107,821	12	5,600	79,827
MADS	P3	0.125	0	95,716	74	6,314	64,247
MADS	P3	0.250	0	97,158	0	6,272	65,824
MADS	P3	0.500	9	107,897	5	5,810	78,818
IDI	P1	0.125	0	93,553	23	5,040	68,369
IDI	P1	0.250	0	96,090	20	5,222	69,905
IDI	P1	0.500	3	107,001	6	4,354	85,081
IDI	P2	0.125	0	93,503	23	5,446	66,329
IDI	P2	0.250	0	96,097	14	5,194	70,135
IDI	P2	0.500	3	107,104	5	4,256	85,616
IDI	P3	0.125	0	93,491	13	4,886	69,096
IDI	P3	0.250	0	96,160	10	4,718	72,518
IDI	P3	0.500	3	107,066	10	3,836	87,845

plan of MADS. Note that we do not include the cost associated with NS in Obj to prevent it from being dominated by this cost.

First, we compare MADS with IDI. Table 5 suggests that MADS improves the route similarity at the expense of time spent. Another general comment is that high values of σ_s result in customers who could not be served in the solution. In such cases, IDI performs better in covering customers than MADS because there is no effort done in creating similarity in scenarios, which provides flexible schedules to serve more customers.

When we analyze Table 5, in particular, for MADS, we see that, in general, increasing the probability of occurrence for a given standard deviation increases time spent. However, there is no general trend for the effect of the probability of occurrence on similarity. In most cases, increasing the standard deviation for a given probability of occurrence increases time spent, and decreases similarity, with the addition that high values result in unserved customers, making the objective value worse. This is expected because increased and dispersed service times make the problem worse with respect to time spent and similarity.

Last, when we analyze Table 5, in particular, for IDI, we find that, in general, increasing the standard deviation for a given probability of occurrence increases the objective value, again because increased and dispersed service times increases time dramatically and decreases similarity.

A set of experiments is done to explore the sensitivity of α_5. The results are shown in Table 6. From the table, we can see that as α_5 increases from 5, the solution is slightly worse. The reason is that in the tabu search, the solution is trapped in local optima early because of the large value of α_5. As α_5 decreases from 5, the solution degenerates quickly. When $\alpha_5 = 0.025$, the solution is similar to the solution of IDI (the row for IDI with P2 and $\sigma_s = 0.250$ in Table 5). When $\alpha_5 = 0$, the solution is different from IDI because the initial daily schedules are constructed differently.

Table 6 MADS Sensitivity to the Weight of Similarity α_5

α_5	NS	Time	Penalty	Sim
0	0	96,330	0	4,718
0.025	0	96,164	1	5,250
1.000	0	96,796	0	6,258
2.500	0	97,307	0	6,328
5.000	0	97,108	0	6,426
7.500	0	97,212	0	6,421
INF	0	97,244	0	6,414

Note. Customers follow distribution P2; standard deviation of service times $\sigma_s = 0.250$.

4.3. MADS vs. Real-Life Solution

In this second set of experiments, we both explore the effect of changing parameters of the MADS algorithm and compare the quality of its solution with the current practice. We run the experiments for both data sets, and we explore the following range of percent buffer capacity: 0%, 20%, 40%, 60%, 80%, 100%. For data set 1, the first half of the real-life data (15 days) is treated as past realizations for the MADS algorithm and the second half (14 days) as future outcomes to evaluate and compare the solutions. For data set 2, similarly, the first 21 days of the real-life data are used to train the master plan, and the remaining 21 days are used to evaluate the solutions.

Tables 7 and 8 show the solutions on the real-life data instances for different buffer capacities of our heuristic for data set 1 and data set 2, respectively. The new heading is BF for the percent buffer capacity. We can omit the column for unserved customers, because in both data sets, all the customers can be feasibly served in each day of the planning horizon. In addition, we provide the solution obtained by the IDI algorithm and the real-life solution. The real-life solution is obtained by a proprietary state-of-the-art routing algorithm of a courier delivery company, and the algorithm is based on territory planning. Routes are planned according to predefined service territories. Each service territory corresponds to a single driver's route. The service territories may be modified and adjusted daily to accommodate fluctuations in drivers' work load because of the varied package volume.

From the results, we can see that the solutions obtained by MADS are better than the real-life solution and the solution by IDI in objective value. Compared with IDI, MADS increases the similarity at the expense of increased time spent. Compared with the current practice, the best solution obtained by MADS is better in all measures for both data sets. This suggests that our heuristic can be tuned to provide improvements over the current practice. We believe that one advantage of MADS is that it does not constrain a route to be within a certain territory. As a result, a route may cross several territories, which provides more flexibility. When we analyze the effect of

Table 7 Comparison of MADS with Real-Life Solution for Data Set 1

Alg	BF	Time	Penalty	Sim	Obj
Real life	—	109,010	5,092	4,116	93,522
IDI	—	97,927	94	2,520	85,421
MADS	0	98,739	57	4,694	75,325
MADS	20	98,635	13	4,702	75,138
MADS	40	98,026	55	4,689	74,636
MADS	60	98,727	1	4,944	75,017
MADS	80	99,873	255	4,640	76,928
MADS	100	98,549	6	3,409	81,511

Table 8 Comparison of MADS with Real-Life Solution for Data Set 2

Alg	BF	Time	Penalty	Sim	Obj
Real-life	—	195,795	16,994	9,114	167,219
IDI	—	192,060	12	7,728	153,432
MADS	0	195,503	4	10,147	144,772
MADS	20	195,452	11	9,990	145,513
MADS	40	195,861	3	10,347	144,129
MADS	60	195,545	5	10,628	142,410
MADS	80	195,036	3	10,493	142,574
MADS	100	195,106	4	12,027	134,975

the buffer capacity, we can see that the solution is best when BF = 40% for data set 1, and when BF = 100% for data set 2. The difference in the solutions is obvious when BF is different. BF provides a flexible tool that can be adjusted to provide good solution. One should try different values of BF and select the best solution. Note that the similarity measure of IDI and real-life solution is based on the master plan of the best solution of MADS, i.e., the master plan of BF = 40% for data set 1 and the master plan of BF = 100% for data set 2.

Last, when it comes to the similarity measure, it is possible to derive an upper bound: the total number of customers occurring in the scenarios that are used to evaluate the solution. For data set 1, the similarity measure of the best solution is 4,689, and the total number of customers is 6,489; therefore the ratio of the similarity measure over the upper bound is $4{,}689/6{,}489 = 72\%$. Similarly, for data set 2, the ratio is $12{,}027/12{,}798 = 94\%$. It means 72% and 94% of the customers are within 0.1 mile of their assigned master routes for data sets 1 and 2, respectively.

4.4. MADS vs. ConVRP Solution

We run MADS over a set of ConVRP benchmark problems (Groër, Golden, and Wasil 2008) to get a solution with consistent routes, and compare it with the solution of ConRTR. We choose data set 2, which is available at: http://www.rhsmith.umd.edu/faculty/bgolden/vrp_data.htm. We have to make some adjustments to MADS because ConVRP has different constraints than our problem, and the benchmark problems are generated differently. ConVRP does not consider time windows, but it requires a customer to be always served by the same vehicle, the precedence constraints are satisfied (if customers i and j are both served by the same vehicle on a specific day and i is served before j, then customer i must be served before j by the same vehicle on all days that they both require service), and service time difference is minimized. In the benchmark problems, all customers have a probability of 0.7 of occurring in a day. Each instance has five days of data. In the master plan, the initial set of customers C^0 includes all customers who occur two or more times in the five days. We use C^0 to train the master plan, and use all five days to evaluate the master plan. It turns out that all customers in C^0 can be scheduled in the master plan. For each day, we simply drop nonoccurring customers, and insert customers who are not in the master plan. The tabu search is run for the master plan, but is skipped for daily routes. In this way, in the final solution, a customer is always visited by the same vehicle, and all the precedence constraints are satisfied. The results are shown in Table 9. The column "Problem" in the table indicates the problem instance number; "Node number" is the total number of customers; "Time" is the total travel time; "Avg" is the average arrival time difference; and "Max" is the maximum arrival time difference. The last row "Average" shows the average result of the 12 instances. We can see that the average result of our solution is better than ConRTR in all three measures. We believe that there are two reasons: (1) the buffer capacity can be tuned to provide different solutions from which we can choose the best and (2) the tabu search is very effective in improving the master plan.

Table 9 Comparison of MADS with ConRTR for a ConVRP Data Set

		ConRTR			MADS		
Problem	Node number	Time	Avg	Max	Time	Avg	Max
1	50	2,282.14	8.36	24.38	2,281.05	6.15	24.27
2	75	3,872.86	6.85	34.26	3,954.99	4.90	30.00
3	100	3,626.22	8.21	22.87	3,636.66	8.79	43.24
4	150	4,952.91	4.93	27.53	4,993.26	4.15	29.73
5	199	6,416.77	3.32	26.93	6,370.41	3.59	37.15
6	50	4,084.24	19.19	63.47	3,967.06	19.54	71.58
7	75	7,126.07	14.19	83.96	7,062.52	16.25	60.88
8	100	7,456.19	22.70	73.04	7,461.98	20.53	67.49
9	150	11,033.54	22.19	106.43	10,872.44	20.10	74.87
10	199	13,916.80	18.47	60.17	13,646.84	18.98	59.48
11	120	4,753.89	4.78	16.10	4,899.94	6.25	23.69
12	100	3,861.35	3.00	17.58	3,938.11	3.59	16.82
Average	—	6,115.25	11.35	46.39	6,090.44	11.07	44.93

5. Conclusions

In this study, we consider a real-life CDP, a variant of the VRPTW with uncertainty in customer occurrence and service times. We present a problem formulation and develop an efficient two-phase heuristic based on insertion and tabu search. Our model represents the uncertainty in service times using robust optimization and the probabilistic nature of customers using scenario-based stochastic programming with recourse. Thus we benefit from the simplicity of a robust model and the flexibility of recourse actions.

We first adapt a nominal VRPTW model for the CDP. We then define a problem-specific recourse action of partial rescheduling of routes by omitting nonoccurring customers and rescheduling new customers. Our model includes a master plan problem, which represents the uncertainty in service times using robust optimization (WC service times), and the subset of possible customers ML to appear. The master plan routes created take into account the similarity with the daily schedules to serve a given number of scenarios. To solve large-sized instances of this CDP model, we develop a two-phase heuristic, MADS, based on insertion. The daily schedules that are obtained from the master plan are improved using a tabu search algorithm.

We explore experimentally the sensitivity of our heuristic to uncertain problem parameters as well as to some control parameters. We also compare the quality of the solution with an IDI algorithm, which does not provide a master plan, and to an industry standard solution, obtained using a territory planning method. We observe that the MADS heuristic improves, in general, the similarity measure at the expense of increased time spent and that it is possible to outperform the current industry practice in all measures. We obtain consistent routes with a slightly modified MADS, and compare them with the solution of ConRTR over a set of ConVRP benchmark problems, and the average result of our solution is better than that of ConRTR.

Acknowledgments

The authors thank UPS for providing the real-life problem data. The authors also thank three anonymous referees and an associate editor for their insightful comments and suggestions. Research was supported by the NSF under Grant CMS-0409887 and by METRANS under Grant 06-11.

References

Beasley, J. E., N. Christofides. 1997. Vehicle routing with a sparse feasibility graph. *Eur. J. Oper. Res.* **98** 499–511.

Ben-Tal, A., A. Nemirovski. 1998. Robust convex optimization. *Math. Oper. Res.* **23**(4) 769–805.

Ben-Tal, A., A. Nemirovski. 1999. Robust solutions to uncertain programs. *Oper. Res. Lett.* **25** 1–13.

Ben-Tal, A., A. Goryashko, E. Guslitzer, A. Nemirovski. 2004. Adjustable robust solutions of uncertain linear programs. *Math. Programming* **99** 351–376.

Bertsimas, D. 1992. A vehicle routing problem with stochastic demand. *Oper. Res.* **40**(3) 574–585.

Bertsimas, D., M. Sim. 2003. Robust discrete optimization and network flows. *Math. Programming* **98** 49–71.

Bertsimas, D., M. Sim. 2004. The price of robustness. *Oper. Res.* **52**(1) 35–53.

Bertsimas, D., P. Jaillet, A. R. Odoni. 1990. A priori optimization. *Oper. Res.* **38**(6) 1019–1033.

Birge, J. R., F. Louveaux. 1997. *Introduction to Stochastic Programming*. Springer-Verlag, New York.

Cordeau, J.-F., G. Laporte. 2004. Tabu search heuristics for the vehicle routing problem. C. Rego, B. Alidaee, eds. *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search*. Kluwer, Boston, 145–163.

Dessouky, M., R. Hall, A. Nowroozi, K. Mourikas. 1999. Bus dispatching at timed transfer transit stations using bus tracking technology. *Transportation Res. Part C* **7** 187–208.

El-Ghaoui, L., F. Oustry, H. Lebret. 1998. Robust solutions to uncertain semidefinite programs. *SIAM J. Optim.* **9** 33–52.

Goldfarb, D., G. Iyengar. 2003. Robust portfolio selection problems. *Math. Oper. Res.* **28**(1) 1–38.

Groër, C., B. Golden, E. Wasil. 2009. The consistent vehicle routing problem. *Manufacturing Service Oper. Management* **11**(4) 630–643.

Haughton, M. 2000. Quantifying the benefits of route reoptimization under stochastic customer demands. *J. Oper. Res. Soc.* **51** 320–322.

Haughton, M., A. Stenger. 1998. Modeling the customer service performance of fixed-routes delivery systems under stochastic demands. *J. Bus. Logist.* **9** 155–172.

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

Jaillet, P. 1988. A priori solution of a traveling salesman problem in which a random subset of the customers are visited. *Oper. Res.* **36**(6) 929–936.

Jula, H., M. M. Dessouky, P. Ioannou. 2006. Truck route planning in non-stationary stochastic networks with time-windows at customer locations. *IEEE Trans. Intelligent Transportation Systems* **37** 51–63.

Lambert, V., G. Laporte, F. Louveaux. 1993. Designing collection routes through bank branches. *Comput. Oper. Res.* **20** 783–791.

Laporte, G., F. Louveaux, H. Mercure. 1992. The vehicle routing problem with stochastic travel times. *Transportation Sci.* **26**(3) 161–170.

Malandraki, C., D. Zaret, J. Perez, C. Holland. 2001. Industrial engineering applications in transportation. G. Salvendy, eds. *Handbook of Industrial Engineering*, 3rd ed. John Wiley and Sons, New York, 787–824.

Morales, J. C. 2006. Planning robust freight transportation operations. Ph.D. thesis, Georgia Institute of Technology.

Savelsbergh, M. W. P., M. Goetschalckx. 1995. A Comparison of the efficiency of fixed versus variable vehicle routes. *J. Bus. Logist.* **16** 163–187.

Sungur, I. 2009. *The Robust Vehicle Routing Problem: A Novel Approach to Uncertainty in a Classical Transportation Problem with Real Life Application*. VDM Verlag Dr. Müller, Saarbrücken, Germany.

Sungur, I., F. Ordóñez, M. M. Dessouky. 2008. A robust optimization approach for the capacitated vehicle routing problem with demand uncertainty. *IIE Trans.* **40** 509–523.

Waters, C. D. J. 1989. Vehicle-scheduling problems with uncertainty and omitted customers. *J. Oper. Res. Soc.* **40** 1099–1108.

Zhong, H., R. W. Hall, M. M. Dessouky. 2007. Territory planning and vehicle dispatching with driver learning. *Transportation Sci.* **41**(1) 74–89.