

Routing Courier Delivery Services with Urgent Demand

Chen Wang, Maged Dessouky
Daniel J. Epstein Department of Industrial and Systems Engineering
University of Southern California
Los Angeles, CA 90089-0193
wang20,maged@usc.edu

Fernando Ordóñez
Industrial Engineering Department
Universidad de Chile
República 701, Santiago, Chile
fordon@dii.uchile.cl

Abstract

Courier delivery services in many cases are faced with sporadic, tightly constrained, urgent requests in addition to regular demand. An example of such an application is the transportation of medical specimens, where an efficient delivery is crucial in providing high quality and affordable service. However, the presence of random urgent requests, due to medical emergencies, can create substantial additional costs if not taken into account.

We model this problem as a multi-trip vehicle routing problem with time windows using stochastic programming with recourse to represent the random urgent requests. We develop an insertion based heuristic with a tabu-search improvement phase to solve this problem. Our computational results show that this approach obtains significant improvement in travel costs as well as in route similarity over alternative methods, both on randomly generated data as well as on a real data set provided by a leading healthcare provider in Southern California.

Keywords: Stochastic vehicle routing; Multi-trip; Time windows; Heuristics

1. Introduction

In this work, we consider a stochastic vehicle routing problem that is faced with urgent random demand. By urgent demand we mean random requests that impose such tight service constraints to the vehicle routing system that they are either outsourced or lead to expensive routing solutions. The courier delivery industry is regularly faced with the challenges posed by urgent demand. Couriers, in addition to meeting an uncertain demand, sporadically receive tightly constrained requests (such as an overnight delivery to an overseas destination or a same day delivery in a large urban area) which challenge the transportation capabilities and could be provided at a premium.

One example of a routing system with urgent demand is the transportation of clinical specimens, which is pervasive in the healthcare industry. On a daily basis, millions of specimens are delivered in the United States from dispersed hospitals and clinics to centralized laboratories for testing. Timely and efficient transportation of specimens is crucial in providing high-quality and affordable patient service in the healthcare industry. The current situation, however, is far from ideal, where lost or delayed delivery of specimen is the most common problem jeopardizing patient safety (Astion et al., 2003). In addition, the cost of transporting clinical specimens is a significant burden to healthcare systems, especially for urgent cases which require prompt courier services. Furthermore, although there is significant research in routing systems with uncertain demand, the scheduling of urgent delivery of medical specimens is still a manual process in practice. This process leads to solutions where the own fleet is used to service routine (possibly random) requests and most urgent requests are outsourced to a courier service, such as taxis. This can be an inefficient solution since the outsourcing cost can be rather high.

Motivated by this medical specimen courier delivery problem, we will consider a multi-trip routing problem where random demand is transported from many distributed locations (hospitals/clinics) to the depot (central laboratory) within given time windows. A random urgent request will be a trip request that has a substantially tighter time window. The hypothesis to be validated with this work is that a routing solution with additional capacity that can accommodate the random urgent requests will be a more

efficient solution. The key question is how to integrate the uncertain urgent demands into the delivery schedule for the routine demands.

There are a number of stochastic vehicle routing approaches in the literature that can be used to take into account the demand uncertainty in this problem. However using approaches such as robust optimization or chance-constrained optimization directly to address the rare urgent requests, will either ignore the unlikely requests, lead to a high cost solution or to prohibitively large problems, such problems are the topic of ongoing research (Shortle et al., 2014; Barrera et al., 2014). Instead, in this work we model the problem as a two stage stochastic optimization problem with recourse. This approach requires considering different scenarios, leading to a large scale routing problem. The overall idea is to understand whether we can sacrifice some optimality with regards to regular demand to free some capacity that will give more flexible routes, which could accommodate the more urgent requests at a lower cost.

Current industry practice runs fixed routes during a planning horizon outsourcing most random requests. This solution, which tends to be expensive, exhibits however a nice feature for practical implementation: that it maintains similar routes for regular requests with customers visited by the same vehicle at roughly the same time every day. Such stability is desirable in repeating systems where the quality of service is important (Groër et al., 2009; Sungur et al., 2010). In this work we take into account route similarity as a desirable measure of a routing solution. The motivation is that a stochastic programming solution with recourse will construct a first stage solution that is adapted to accommodate each scenario. But, to provide a high quality of service it is preferable to have a first stage solution that does not change much on each scenario.

To summarize, we propose a model and solution algorithm for the vehicle routing problem with urgent requests, inspired by the healthcare delivery application. We formulate this problem as a multi-trip vehicle routing problem over a given planning horizon, with uncertain demand that occurs continuously in the planning horizon, some of which is urgent and has tight time windows. The stochastic programming problem considers objectives that aim to minimize the outsourcing cost, the route length and the dissimilarity between the first stage solution and the recourse actions taken in each scenario, which aims to improve the service quality. We consider a weighted

combination of these objectives. Our computational studies investigate the influence of these parameters on the different cost measures of the solution obtained and help provide solutions that provide an efficient tradeoff between them. The proposed heuristic solution algorithm builds a first stage solution, or master route, that will require little modification when adapted for each uncertainty scenario by limiting the recourse actions considered.

The rest of the paper is organized as follows. In section 2, a literature review of the relevant problems is presented. Section 3 introduces the problem formulation. In section 4, we present our heuristic solution technique for the problem. Computational results are presented and discussed in section 5. The computational results are separated in experiments of the proposed heuristic on randomly generated data sets and experiments on a real data set from a large healthcare provider in Southern California. We summarize the main conclusions of the paper in section 6.

2. Literature Review

The VRP variants related to this work include multi-trip VRP (MVRP) and stochastic VRP (SVRP). Of the vast literature in SVRP our work is related to stochastic programming approaches for problems under uncertainty. Another relevant notion is the idea that route similarity is related to customer service quality in the vehicle routing problem. We next present a short summary of the prior work in these areas.

Multi-trip VRP (MVRP), as a variant of the VRP, has gained little attention in the literature. In the MVRP, vehicles can be used more than once during the planning horizon. Taillard et al. (1996) suggest in their study that assigning more routes to a vehicle is a more practical solution in real life. Brandao and Mercer (1997, 1998) extended the study on multi-trip VRP by also including the delivery time window and the capacity of the vehicles. Petch and Salhi (2003) integrate the approaches proposed by Taillard et al. (1996) and Brandao and Mercer (1997, 1998). Azi et al. (2006) first describe an exact algorithm for solving a multi-trip VRP problem of one vehicle with time windows. Salhi and Petch (2007) provide a comprehensive literature review on the

multi-trip VRP, and present a genetic algorithm based on a heuristic for the solution of MVRP. In recent years, Zapfel and Bogla (2008) provide a study of a multi-trip vehicle routing and crew scheduling with overtime and outsourcing options. Ren et al. (2010) introduce the use of shifts into the VRP, and study a new variant of the VRP, which is with time windows, multi-shifts, and overtime. Martinez and Amaya (2013) consider a multi-trip problem with time windows which integrates the packing of circular items in the vehicle. Our work builds on this previous literature by using multi-trip routes as efficient recourse strategies in a stochastic VRP problem with customer uncertainty.

Another variant of the multi-trip VRP is the periodic VRP (PVRP), which customers have to be visited once or several times in the planning horizon (Angelelli and Speranza, 2002). PVRP extends the planning horizon being considered to plan routes over several days. Angelelli and Speranza (2002) propose a Tabu search based heuristic for the solution of a PVRP with intermediate facilities. Francis and Smilowitz (2006) present a continuous approximation for service choice of a PVRP with capacity constraints. Hemmelmayr et al. (2009) propose a new heuristic for solving PVRP as well as a Periodic Travelling Salesman Problem, based on a neighborhood search. The paper of Alonso et al. (2008) extends the classical VRP to a periodic and multi-trip VRP with site-dependency and proposes a Tabu search based algorithm.

The stochastic vehicle routing problem (SVRP) introduces uncertainty in the parameters. Ichoua et al. (2006) reviews the literature in SVRP and classifies the SVRP into two subgroups of problems: static stochastic vehicle routing problems (SSVRP) and dynamic stochastic vehicle routing problem (DSVRP). In the SSVRP, the customers and/or demands are random variables. The vehicle routing problem with stochastic demands (VRPSD) (Yang et al., 2000), the vehicle routing with stochastic customers (VRPSC) (Waters, 1989), the vehicle routing problem with stochastic customers and demands (VRPSCD) (Gendreau et al., 1995), and the probabilistic travelling salesman problem (PTSP) (Laporte et al., 1994) belong to SSVRP. One typical solution technique for the SSVRP is the two-stage method (Gendreau et al., 1996; Bertsimas and Simchi-Levi, 1996), where in the first stage, an “a-priori sequence” solution (Bertsimas et al., 1990) is proposed, and in the second stage, recourse actions (e.g., skipping non-occurring customers, returning to the depot when capacity is exceeded, or complete rescheduling

for occurring customers) is allowed to adjust an “a-priori solution” after the uncertainty is revealed. Another solution technique for the SSVRP is the “re-optimization” approach (Secomandi, 2001; Novoa and Storer, 2009), where dynamic programming solutions are developed.

DSVRP studies the problems where new events occur over time and no “a-priori” solution is utilized. There are two different ways to exploit the probability information in the literature: analytical studies and stochastic algorithms. The analytical studies provide new insights to the solution structure, thus helping to design more efficient deterministic algorithms (Bertsimas and Simchi-Levi, 1996). For problems with uncertainty (e.g., Spivey and Powell, 2004), researchers have been studying stochastic and dynamic algorithmic approaches that include current information and future probabilistic events to produce more efficient solutions. A recent comprehensive review of DSVRP appears in Pillac et al. (2013). To the best of our knowledge there is no work on DSVRP which makes use of the flexibility of multi-trip routes.

The healthcare courier delivery problem has a high requirement on the quality of customer service. Some recent work has included customer service in the models for fixed route delivery systems under stochastic demand (Haughton and Stenger, 1998). Haughton (2000) develops a framework for quantifying the benefits of route re-optimization, also under stochastic customer demands. Zhong et al. (2007) propose an efficient way of designing driver service territories, considering uncertainty in customer locations and demand. Groër et al. (2009) introduce the Consistent VRP (ConVRP) model, with an objective of obtaining consistent routes such that the customers are visited by the same driver at roughly the same time on each day. Tarantilis et al. (2012) propose a solution approach for the ConVRP that is based on a template solution and implements a tabu search. Sungur et al. (2010) introduce the concept of “route similarity” as the number of customers of the daily routes that are within a given distance of any customer on the master route, and use it as a key measure for developing optimal routing strategies.

Our proposed model is most similar to the models proposed by Groër et al. (2009) and Sungur et al. (2010) in that route similarity is a key aspect of the routing problem. It differs from this prior work however because the nature of the urgent requests may require a sudden reroute to the depot. This forces a novel multi-trip VRP formulation

with variable number of trips per vehicle and allows considering more general recourse actions than what is considered in these two prior references.

3. Vehicle Routing with Urgent Requests

We formulate a multi-trip vehicle routing model for the healthcare industry courier delivery problem. In this multi-trip model we allow vehicles to optimize the length and the number of multiple trips to adjust to the demand scenario. For example, a MVRP may require the customers to be visited twice in two trips in a workday, with a fixed trip length. A PVRP may have all the customers be visited in one trip each workday during the planning horizon, where the length of a trip of a vehicle is 8 hours each day. In our problem, the vehicles operate in multiple trips during the planning horizon where the number of trips and the lengths of each trip are not defined a-priori, but adjust to the demand scenario being serviced. In this section, we provide a mixed integer linear programming formulation of this multi-trip VRPTW with stochastic clients.

Assume we are making a routing schedule for a healthcare courier delivery service provider. There are n potential customers (hospitals, clinics) in the region that must be visited during a planning horizon by a fleet of identical vehicles. Without loss of generality in what follows we will assume that the planning horizon will be a day. Each request for service has a location, pick up time window and a latest drop-off time (or drop-off deadline). The locations and time windows of all the potential customers are known ahead of time, however, which customers have requests on a specific instance is only revealed at the start of the planning horizon (day). This uncertainty is represented by a set of scenarios $\{1, \dots, \delta\}$, with scenario d occurring with a given probability p_d . We seek to obtain a master route, using a subset of the total customers that will be referred to as scenario $d=0$. This master route is used to measure the consistency of the routes that satisfy the demand realized in each scenario. The question of which locations to consider for the master schedule will be explored in the computational results. There is one depot (node 0) located at the central lab. Each vehicle should leave the depot/lab at the beginning of the planning horizon (day), and return to the depot/lab at the end of the day. It can also return to the depot/lab anytime during the day when required (i.e., when there

are urgent requests that need samples delivered by a certain time at the lab). As each vehicle has multiple trips, we assume a dummy depot (represented by node $n + 1$) located also at the central depot/lab to keep track of which trip the request is on. We keep track of any unmet request, which we assume is outsourced. For brevity, we will use the term “taxi” to refer to an outsourced vehicle for the remainder of the paper. The notation of the model is as follows.

The routing parameters:

- D : set of scenarios $D = \{0, 1, \dots, \delta\}$.
- C : set of customers, $C = \{1, \dots, n\}$.
- K : set of vehicles.
- W : set of daily trips of a vehicle, $W = \{1, \dots, n\}$.

The cost parameters:

- t_{ij} : travel time between node i and j .
- α_t : unit travel cost, dollars per unit time.
- α_{oi} : unit outsource cost, dollars per taxi trip to service node i .
- α_s : unit dissimilarity cost, dollars for each count of dissimilarity.

The stochastic parameters:

- p_d : probability of occurrence of scenario d .
- C^d : set of occurring customer requests on scenario d .
- s_i^d : service time of customer request i on scenario d .
- a_i^d : earliest time that customer request i can be visited on scenario d .
- b_i^d : latest time that customer request i can be visited on scenario d .
- l_i^d : latest time that customer request i can be delivered to the lab on scenario d .

Other parameters:

- M : a sufficiently large number.

The routing variables:

$$x_{ijk}^d = \begin{cases} 1, & \text{if vehicle } k \text{ travels from node } i \text{ to } j \text{ on scenario } d \\ 0, & \text{otherwise} \end{cases}$$

$$x_{0ik}^{wd} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from the depot to customer } i \text{ on trip } w \text{ on scenario } d \\ 0, & \text{otherwise} \end{cases}$$

$$x_{i(n+1)k}^{wd} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from customer } i \text{ to the lab on trip } w \text{ on scenario } d \\ 0, & \text{otherwise} \end{cases}$$

y_{ik}^d : the time vehicle k arrives at customer i on scenario d .

y_{0k}^{wd} : the time vehicle k leaves the depot for its trip w on scenario d .

$y_{(n+1)k}^{wd}$: the time vehicle k returns to depot from its trip w on scenario d .

The auxiliary demand variables:

$$z_{ik}^{wd} = \begin{cases} 1, & \text{if vehicle } k \text{ visits customer } i \text{ on trip } w \text{ on scenario } d \\ 0, & \text{otherwise} \end{cases}$$

$$u_i^d = \begin{cases} 1, & \text{if customer } i \text{ is visited by a taxi on scenario } d \\ 0, & \text{otherwise} \end{cases}$$

$$r_{ik}^d = \begin{cases} 1, & \text{if vehicle } k \text{ visits customer } i \text{ either on scenario } d \text{ or } 0, \text{ but not both} \\ 0, & \text{otherwise} \end{cases}$$

Before the mathematical formulation of the model is presented, some clarification on the parameters and decision variables need to be made.

- 1) The model considers a planning horizon of one day with $\delta=|D|-1$ demand scenarios to represent the uncertainty. Scenario $d = 0$ is formed with the data that defines the base case that is used to build the master route. We refer to the solution in $d = 0$ as the master route and it is the route with respect to which the similarity of each scenario d routes are measured.
- 2) The maximum number of trips each vehicle can make in a day is n . We allow artificial trips that do not deal with any customers, but just “move” from the depot/lab to the depot/lab without spending any actual time.
- 3) The travel time between node i and j is t_{ij} . In particular, t_{0i} is travel time between the depot/lab and node i ; $t_{i(n+1)}$ the travel time between node i and depot/lab.
- 4) r_{ik}^d is defined as the measure of dissimilarity, with mathematical expression $r_{ik}^d = |\sum_{w \in W} z_{ik}^{wd} - \sum_{w \in W} z_{ik}^{w0}|$. Variable r_{ik}^d equals 1 only if customer i is visited by vehicle k either on scenario d or the master route, but not both. In other

words the dissimilarity counts if a customer is visited by a different vehicle than in the master route.

Problem formulation:

Minimize

$$\begin{aligned} & \alpha_t \cdot \sum_{d \in D} p_d \sum_{k \in K} \left(\sum_{i \in C^d} \sum_{j \in C^d, i \neq j} t_{ij} \cdot x_{ijk}^d + \sum_{w \in W} \sum_{i \in C^d} t_{0i} \cdot x_{0ik}^{wd} + \sum_{w \in W} \sum_{i \in C^d} t_{i(n+1)} \cdot x_{i(n+1)k}^{wd} \right) \\ & + \sum_{d \in D} p_d \sum_{i \in C^d} \alpha_{oi} \cdot u_i^d + \alpha_s \cdot \sum_{d \in D \setminus \{0\}} p_d \sum_{k \in K} \sum_{i \in C^d} r_{ik}^d \end{aligned} \quad (3.1)$$

Subject to:

Routing constraints:

$$\sum_{k \in K} \sum_{j \in C^d, j \neq i} x_{jik}^d + \sum_{k \in K} \sum_{w \in W} x_{0ik}^{wd} + u_i^d = 1 \quad i \in C^d, d \in D \quad (3.2)$$

$$\sum_{w \in W} x_{i(n+1)k}^{wd} + \sum_{j \in C^d, j \neq i} x_{ijk}^d = \sum_{w \in W} x_{0ik}^{wd} + \sum_{j \in C^d, j \neq i} x_{jik}^d = \sum_{w \in W} z_{ik}^{wd} \quad k \in K, i \in C^d, d \in D \quad (3.3)$$

$$\sum_{i \in C^d} x_{0ik}^{wd} = \sum_{i \in C^d} x_{i(n+1)k}^{wd} \leq 1 \quad k \in K, w \in W, d \in D \quad (3.4)$$

$$\sum_{i \in C^d} x_{0ik}^{wd} \geq \sum_{i \in C^d} x_{i(n+1)k}^{(w+1)d} \quad k \in K, w \in W, d \in D \quad (3.5)$$

$$y_{ik}^d + t_{ij} + s_i^d \leq y_{jk}^d + M \cdot (1 - x_{ijk}^d) \quad i \in C^d, j \in C^d, i \neq j, k \in K, d \in D \quad (3.6)$$

$$y_{0k}^{wd} + t_{0j} \leq y_{jk}^d + M \cdot (1 - x_{0jk}^{wd}) \quad j \in C^d, w \in W, d \in D, k \in K \quad (3.7)$$

$$y_{ik}^d + t_{i(n+1)} + s_i^d \leq y_{(n+1)k}^{wd} + M \cdot (1 - x_{i(n+1)k}^{wd}) \quad i \in C^d, w \in W, d \in D, k \in K \quad (3.8)$$

$$y_{(n+1)k}^{wd} \leq y_{0k}^{(w+1)d} \quad w \in W, d \in D, k \in K \quad (3.9)$$

$$a_i^d \leq y_{ik}^d \leq b_i^d \quad i \in C^d, k \in K, d \in D \quad (3.10)$$

$$-M \cdot (1 - z_{ik}^{wd}) + y_{0k}^{wd} \leq y_{ik}^d \leq y_{(n+1)k}^{wd} + M \cdot (1 - z_{ik}^{wd}) \quad i \in C^d, w \in W, d \in D, k \in K \quad (3.11)$$

$$y_{(n+1)k}^{wd} \leq l_i^d + M \cdot (1 - z_{ik}^{wd}) \quad i \in C^d, w \in W, d \in D, k \in K \quad (3.12)$$

$$-r_{ik}^d \leq \sum_{w \in W} z_{ik}^{wd} - \sum_{w \in W} z_{ik}^{w0} \leq r_{ik}^d \quad i \in C^d, k \in K, d \in D \setminus \{0\} \quad (3.13)$$

Domain constraints:

$$x_{ijk}^d \in \{0,1\}, \quad i, j \in C^d, k \in K, d \in D \quad (3.14)$$

$$x_{0ik}^{wd} \in \{0,1\}, \quad i \in C^d, w \in W, k \in K, d \in D \quad (3.15)$$

$$x_{i(n+1)k}^{wd} \in \{0,1\}, \quad i \in C^d, w \in W, k \in K, d \in D \quad (3.16)$$

$$y_{ik}^d \geq 0, \quad i \in V^d, k \in K, d \in D \quad (3.17)$$

$$y_{0k}^{wd} \geq 0, \quad w \in W, k \in K, d \in D \quad (3.18)$$

$$y_{(n+1)k}^{wd} \geq 0, \quad w \in W, k \in K, d \in D \quad (3.19)$$

$$z_{ik}^{wd} \in \{0,1\}, \quad i \in C^d, w \in W, k \in K, d \in D \quad (3.20)$$

$$r_{ik}^d \geq 0, \quad i \in C^d, k \in K, d \in D \quad (3.21)$$

$$u_i^d \in \{0,1\}, \quad i \in C^d, d \in D \quad (3.22)$$

As previously described, the healthcare courier delivery problem should focus not only on plans with minimum travelling cost, but also those with high level of customer service. Therefore, the objective function of our model, as shown in Equation (3.1), is to minimize the expected total cost, that is composed of traveling cost, outsourcing cost, and route dissimilarity cost. The travel cost is represented by

$$\alpha_t \cdot \sum_{d \in D} p_d \sum_{k \in K} \left(\sum_{i \in C^d} \sum_{j \in C^d, i \neq j} t_{ij} \cdot x_{ijk}^d + \sum_{w \in W} \sum_{i \in C^d} t_{0i} \cdot x_{0ik}^{wd} + \sum_{w \in W} \sum_{i \in C^d} t_{i(n+1)} \cdot x_{i(n+1)k}^{wd} \right),$$

which is proportional to the expected total time traveled by all the vehicles in the planning horizon. Here we take $p_0=1$ so the objective is actually the cost of the master route and the expected travel, taxi and dissimilarity cost over the scenarios. The outsourcing cost is represented by $\sum_{d \in D} p_d \sum_{i \in C^d} \alpha_{oi} \cdot u_i^d$, which corresponds to a cost of α_{oi} if node i has to be serviced by a taxi. This cost can represent a fixed cost per taxi trip and also account for the cost per distance t_{i0} of servicing node i by setting $\alpha_{oi} = \alpha_{of} + \alpha_{ov} t_{i0}$, where α_{of} corresponds to the fixed cost and α_{ov} the per unit distance variable cost. The expected route dissimilarity cost is measured by $\alpha_s \cdot$

$\sum_{d \in D \setminus \{0\}} p_d \sum_{k \in K} \sum_{i \in C^d} r_{ik}^d$, which is proportional to the total number of customers that are serviced by a vehicle different from the one servicing it in the master route.

There are two groups of constraints in our model, namely routing constraints and domain constraints. Constraints (3.2) ensure that on each planning horizon (day) every customer is either visited directly from the depot/lab, or right after a vehicle services customer j , or by a taxi when the regular fleet is unavailable. Constraints (3.3) enforce the vehicle flow constraints and help define variable z_{ik}^{wd} which characterizes whether vehicle k visits location i in trip w of scenario d . This variable is key in enforcing time windows to depot/lab and dissimilarity. Constraints (3.4) ensure that each individual trip should start with leaving the depot/lab and end by returning to the depot/lab. Constraints (3.5) enforce the usage of early trips as much as possible, which force the empty trips close to the end of the day instead of at the beginning of the day. Constraints (3.6) assure the relationship of arrival times at customers i and j , when customer j is visited right after i is visited. Constraints (3.7) express the relationship of arrival time to customer j , when j is the first customer request a vehicle handles in a trip. Constraints (3.8) express the relationship of arrival time to customer j , when j is the last customer request a vehicle handles in a trip. Constraints (3.9) enforce that the finish time of a trip of a vehicle should be no later than the start time of the next trip of the vehicle. Together constraints (3.6)-(3.9) define arrival times of vehicles at the different customers and the multiple visits to the depot/lab. These constraints, which correspond to an adaptation of MTZ constraints to a multi-trip model, eliminate infeasible subtours. In fact MTZ constraints are exactly constraint (3.6), but given the special conditions of depot/lab and multiple trips these constraints for these additional conditions are translated into (3.7)-(3.9). Constraints (3.10) enforce the arrival time of a vehicle at a customer to be in the required time window for handling the customer request. Constraints (3.11) require that the arrival time at a customer on a trip should be between the start time and the end time of the trip. Constraints (3.12) require that each vehicle should visit the lab before the drop-off deadline of each specimen collected by a vehicle on a trip. Constraints (3.13) are another representation of our expression for dissimilarity $r_{ik}^d = |\sum_{w \in W} z_{ik}^{wd} - \sum_{w \in W} z_{ik}^{w0}|$. They remove the usage of the absolute value in the expression, so that the system is linearized. Constraints (3.14)–(3.22) are the variable domain constraints.

The objective function of this model aims to obtain a master route which has a small transportation cost and that it leads to scenarios which balance the transportation cost with the dissimilarity cost. The solution heuristic described in the next section will be built minimizing both the cost of the master route and the expected costs of the scenarios.

One question that remains from the model introduced above is how to construct the master route, in other words, which customers should be considered in scenario $d=0$. The objective is to balance route similarity (tied to higher customer service) and travel time efficiency. This is not straight forward since including very few nodes in the master routes will lead to routes that have little similarity but are efficient, while considering too many nodes in the master will create routes that may be very similar but which will have long travel cost since many locations will not be visited in each scenario. In the computational section we explore computationally how to build master routes to tradeoff travel time cost and route similarity.

4. Heuristic

The problem formulated in the previous section results in a large routing problem that is difficult to solve exactly. Indeed, since there are $|D|$ scenarios that have to be taken into account in the routing, and each vehicle makes n trips during the planning horizon (a day - including real and artificial trips), then solving a problem with n customers and k vehicles is equivalent to solving a routing problem with $n|D|$ customers with $k|D|$ vehicles that can do up to n trips each. Specifically, the real world instance from the healthcare industry that we consider in this paper has $|D| = 30$, $n = 105$ and $k = 14$, which is equivalent to a routing problem with at least $k|D| = 420$ vehicles on $n|D| = 3150$ nodes. Such a problem size is likely to be solved in hours, if at all, rather than minutes with existing exact solution methods. Therefore, we develop heuristic algorithms that will be able to provide solutions in minutes to the route planner. We present the heuristic in four parts: insertion, tabu search, constructing master routes, and constructing operational (or daily) plans. The central idea of the heuristic is to separate the problem for each scenario $d \in D$ and solve various smaller routing problems with

appropriate cost functions. The heuristic begins by constructing a master route, for $d=0$, taking into account only regular vehicle travel cost. Routes for every other scenario $d \in D$, $d \neq 0$ are then constructed starting from the master routes and considering the part of the objective function that is relevant to scenario d . The insertion and tabu search procedures are used to construct efficient routes in both parts of the heuristic (master and daily routes).

This heuristic procedure is inspired by the one developed for a courier delivery problem under uncertainty in Sungur et al. (2010) but has the following important differences with this prior work. The heuristic implements the more versatile recourse actions considered in this problem which allow creating different multi trips on different scenarios. The heuristic proposed here does not consider a phase 2 where information about unmet demand in second stage scenarios is given to re-optimize the master problem as in Sungur et al. (2010). The reasons for not including feedback to the master problem are due to the nature of the urgent VRP problem. In this problem an unmet demand is more often due to the occurrence of urgent requests in a given scenario than this demand being difficult to satisfy. Therefore, including the demand that was unmet in a scenario in the master might not be always beneficial. A possible recourse is to include all urgent requests on the master route to satisfy the unmet demand in the scenario in question. This, however, would generate overly conservative master routes that seek to satisfy even the rare urgent requests. Since this seems to provide too much information to the solution procedure we opted to develop a one shot heuristic without feedback from the scenarios to the master problem.

4.1 Insertion

Insertion heuristics are popular for solving vehicle routing and scheduling problems because they are fast, easy to implement, produce good solutions, and are easily extendable to handle complicating constraints. A comprehensive review of insertion heuristics can be found in Campbell et al. (2004). Our heuristic uses an insertion technique as the building block for constructing routes. The insertion heuristic used for constructing master routes only considers travel time, while insertion for daily routes

considers the complete objective function relevant to each scenario and starts from the master routes.

Algorithm 1 below describes the insertion heuristic for building master routes, while the insertion algorithm for constructing daily routes is presented in Algorithm 1.1.

Algorithm 1: Insertion of request to form master routes

Input: the scheduled routes; a request to insert.

Output: the updated routes and taxi cost.

for all the positions in all the active routes

 Find the feasible insertion positions with minimum insertion cost;

end for

if the insertion is feasible *then*

 Update the routes adding the insertion of minimum insertion cost;

else if there is an idle vehicle *then*

 Put the request on the new vehicle;

else request is serviced by taxi, increment taxi cost;

end if

Algorithm 1.1: Insertion of a daily request not in the master routes

Input: the scheduled routes; the master routes; a request to insert.

Output: the updated routes.

calculate taxi cost of request;

Set minimum insertion cost $2 * \text{taxi cost of request}$;

for all the positions in all routes

 find the feasible insertion position with minimum insertion cost;

 update minimum insertion cost;

end for

if minimum insertion cost is smaller than taxi cost *then*

 use fleet;

```

        update routes;
    else use taxi and update taxi cost;
end if

```

In these insertion algorithms we have to keep track of the arrival to customers and the lab to check if an insertion is feasible. Omitting the indices of scenario, vehicles and trip for simplicity, we can express the start time of visit to node i as follows: $y_i = \max(y_{i-1} + s_i + t_{i-1,i}, a_i)$, where node $i-1$ and node i are the two nodes consecutively visited by a vehicle. The earliest time a vehicle can visit node i is a_i and $t_{i-1,i}$ is the travel time between node $i-1$ and node i .

To check the feasibility of an insertion we verify that both the pickup and drop off at the lab are within bounds. This means that $a_i \leq y_i \leq b_i$ and that the next time (after picking up item i) that the vehicle visits the depot/lab satisfies $y_{n+1} \leq l_i$ for all items i picked up in that trip.

The cost on the time traveled if the pickup is inserted as node $i-1$ and the delivery is inserted as node i in a route (see Figure 4.1), can be calculated as $t_{i-2,i-1} + t_{i-1,i} + t_{i,i+1} - t_{i-2,i+1}$. If the pickup is inserted as node $i-1$ and the delivery is inserted as node $i+a$ ($a \geq 1$) (see Figure 4.2), then the insertion cost can be calculated as $t_{i-2,i-1} + t_{i-1,i} + t_{i+a-1,i+a} + t_{i+a,i+a+1} - t_{i-2,i} - t_{i+a-1,i+a+1}$. The taxi cost is made up of two parts in the algorithms. One is a fixed pickup cost, which is proportional to the number of trips. The other is the variable cost, which is proportional to the travel time from the pickup location to the delivery location. The cost for dissimilarity is calculated by comparing the scheduled routes to the master routes. If a request is serviced by the same vehicle, then the dissimilarity is 0; otherwise, it is 1. It should be noted that we assume the dissimilarity cost is always 1, when a customer is visited by a taxi.

Figure 4.1: Pickup is followed directly by delivery

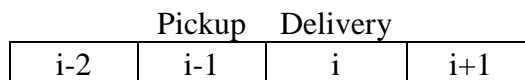
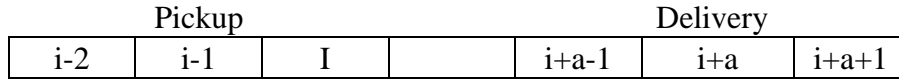


Figure 4.2: Delivery occurs $a+1$ stops after pickup.



The insertion heuristic for master routes is sequential and activates a new vehicle when it is not feasible to handle the request with a currently active vehicle. This approach is favored for its lower usage of vehicles in the master routes, which is another factor of cost reduction for the healthcare provider.

4.2 Tabu Search

Insertion heuristic algorithms are used to build initial solutions for the master and the daily routes, and a Tabu search algorithm (Algorithm 2) is developed as the post phase improvement for efficient master and daily routes. The implementation of the Tabu search considers the neighborhoods obtained from a 2-opt exchange move (Lin, 1965) and a λ -interchange move (Osman, 1993). The λ -interchange operators are generated by randomly selecting two requests from two different routes, and exchanging the requests by interchanging the pickup and the delivery of each request. As the problem requires that the pickup and delivery of a request be handled by the same vehicle, it must be assured that the pickup and the delivery of a request stay on the same vehicle. The 2-opt exchange operator is generated by randomly selecting two nodes (pickup or delivery) from the same randomly selected vehicle. As a specimen can only be delivered after it is picked up, it must be assured that the delivery of any request is located after the pickup of the request.

Algorithm 2: Tabu Search Algorithm

Input: a master route or a daily plan to improve

Output: improved master route or daily plan

repeat

randomly chose two routes from the solution

generate η_{max} neighbors from λ -interchange operator

generate γ_{max} neighbors from 2-opt operator

choose the best solution and make the move;
randomly generate a tabu tenure θ from a uniform distribution $U(\theta_{min}, \theta_{max})$;
if the move is λ -interchange *then*
 set the tabu for moving the exchanged requests for θ iterations;
else
 set the tabu for moving the exchanged nodes for θ iterations;
end if
until no improvement in I_{max} iterations;
calculate the objective and save the current solution;

In each iteration, the Tabu search generates η_{max} λ -interchange neighbors and γ_{max} 2-opt neighbors of the current solution. The number of Tabu iterations θ is a random number uniformly distributed in $(\theta_{min}, \theta_{max})$. The Tabu search at each iteration moves to the best neighbor. A temporary move to a worse solution is allowed to escape from a local minimum. The Tabu status is overridden if the new solution improves from the best solution. The algorithm terminates if there is no improvement in I_{max} iterations.

The Tabu search algorithm is applied to both the master routes and the daily routes. When it is applied to master routes, the objective is to minimize the total time traveled, as to have more slack time to accommodate the random requests. When it is applied on daily routes, the objective is to minimize the cost including total time traveled, taxi cost, and route dissimilarity.

4.3 Master Routes

Master routes must consider the following conflicting objectives: an efficient template for regular demands and flexibility to adapt to the random urgent requests that arise during operations. A customer that requests service regularly usually has wide time windows and should be considered a regular request in the master route. A random urgent request with tight time windows that occurs rarely should not be included in the master route.

Algorithm 3 below describes the method of constructing master routes. The idea is to include the customers that have a high probability of occurrence. The objective is to obtain a solution that is likely to visit many of the customers that appear frequently, thus incurring a small additional cost to adapt to the actual customers that appear on scenario d . This is the way the proposed heuristic brings to the first phase problem information from the uncertain future scenarios of the stochastic programming problem. An insertion algorithm is used to construct an initial solution for the master routes. Tabu search is used to improve the efficiency in travel time so that more slack is obtained for more random urgent requests.

Algorithm 3: Formation of a Master Route

Input: All the customers to insert; the probability of a customer to request service; a threshold for probability of customer occurring

Output: Master routes

for all the customers *do*

if the occurring probability of a customer is larger than the threshold *then*

 include the customer into the master route by calling Algorithm 1;

end if

end for

improve the master routes with Tabu search by calling Algorithm 2;

4.4 Daily Plans with Urgent Requests

As described earlier, in the first stage, we obtain the solution of an effective master route, and in the second stage, we adjust the planned routes to handle the urgent requests. The objective of the second stage is to accommodate as many of the urgent requests as possible with the existing fleet, including the slack time of the vehicles for the master routes. In this second stage, we need to quickly modify the master route to service the updated requests.

If the recourse action allows skipping customers then the problem can be approximated by a *knapsack* problem (Kellerer et al., 2004). The recourse strategy is

inspired by the classic recourse strategy (strategy b) in Bertsimas (1992), which assumes the demand will be revealed before the vehicle leaves the depot to service the customer. Therefore, a customer will be skipped if it does not request service on a particular scenario.

In our strategy, we also make the same assumption that the travel time and the actual demand on each scenario are known before the vehicle departs from the depot/lab. The recourse action on each day includes skipping the customers in the master routes that do not request service from the master route and inserting the customers who request service into the existing routes if possible. The heuristic algorithm for building an operational plan (or daily plan) for specific requests during the planning horizon by adapting the master route using recourse action can be found in Algorithm 4.

Algorithm 4: Formation of Daily Plans

Input: the master route; daily requests

Output: the daily plans

```
for each day (scenario) do
    take the master route (generated by Algorithm 3) as the initial daily plan;
    for all the requests in the master route
        if the request does not occur on the day then
            drop the request from the daily plan;
        end if
    end for
    for all the requests on the day do
        if a request is NOT included in the master route then
            insert the request into the daily plan by calling Algorithm 1.1;
        end if
    end for
    improve the daily plan with Tabu search by calling Algorithm 2;
    for all the requests serviced by taxi do
        try inserting the request into the daily plan again by calling Algorithm 1.1;
    end for
```

5. Experimental Results

We conduct computational results to evaluate the heuristic solutions on randomly generated data and for a problem constructed from a real instance in the health-care industry. The computational results on randomly generated data study the effect of different criteria to build the master route and compare these heuristic solutions with what can be obtained by current industry practice (use taxi for any random request) and with routing each day independently (no route similarity). The results on the real instance test the efficiency of the heuristic against the industry solution for a real instance. The heuristic was programmed in C++ and experiments run on a Dell Workstation with dual Intel Xeon 3.20 GHz processors, a 2 GB RAM, on an Red Hat Enterprise Linux operating system.

Given the size of this multi-trip vehicle routing problem under uncertainty which includes urgent requests, our computational experiments only evaluate the quality of the solutions obtained by the heuristic constructed. The development of exact algorithms or lower-bounds is difficult for such large structured problems, and is the topic of active research in the vehicle routing literature. In particular, since the problem should consider rare urgent requests, it is not possible to do so without considering many uncertainty scenarios, otherwise the uncertain requests will be likely and should be considered in the expected case. Therefore, in this paper we focus only on comparing our proposed heuristic algorithm against alternative heuristics for solving the problem.

5.1 Results on Randomly Generated Data Sets

We first use the heuristic developed on randomly generated data sets to evaluate the impact of constructing master routes. The objective is to compare four possible solution strategies to solve the routing problem when there are a set of regular fixed requests that occur every day and a set of random requests that occur in certain scenarios. In particular we consider the following four strategies:

- A. TAXI: schedule all the deterministic requests as master routes using the insertion/tabu heuristic algorithms; use a third party courier, i.e., taxi, for all the random requests. (Apply Algorithm 3 with a customer occurrence probability threshold of 1 to build the master routes; handle all the random requests by taxi.)
- B. IND: form a schedule independently for each day, using the insertion/tabu heuristic algorithms. (Use Algorithms 1 and 2 to build daily routes independently.)
- C. MFIX: schedule the deterministic requests as master routes, and insert the random requests into the scheduled routes on each day. Use taxi if it is infeasible or more expensive to insert the random request into the scheduled routes. (Use Algorithm 3 to build the master routes with a customer occurrence probability threshold of 1.)
- D. MHALF: schedule the deterministic requests and high occurring probability requests (those who have an occurrence probability of 0.5 or higher) as master routes. In the daily schedules, skip the non-occurring customers and insert the unscheduled random requests into the scheduled routes. Use a taxi if it is infeasible or more expensive to insert the random request into the scheduled routes. (Use Algorithm 3 to build the master routes with a customer occurrence probability threshold of 0.5.)

Strategies MFIX and MHALF correspond to solutions proposed by the model introduced in this paper, with a different number of requests in the master route. TAXI corresponds to the current industry practice strategy, where every random request is sent in a third party courier. Finally IND corresponds to the extreme solution that only minimizes travel cost in each scenario, ignoring route similarity. This solution can serve as an optimal travel cost benchmark. Our results will compare these four strategies according to their average travel time per vehicle, average taxi cost, average route dissimilarity, average number of taxi trips, average travel time per request served, and average total cost. We note that as there are no master routes generated in strategy IND, the dissimilarity is calculated by comparing the daily routes to the master routes generated in strategy MFIX.

To build the random instances, we consider a square city that is from -10 to 10 miles in both the x-axis and the y-axis. The depot and the only lab where all the vehicles

start and end their services every day are located at the center of the city, that is $(0, 0)$ on the two-dimensional plane. The location of all potential customers are generated uniformly in the city. Some customers request service at a fixed time every scenario (regular deterministic requests), while others only request service at a fixed time on some of the scenarios (urgent random requests). Each random request has a probability p of occurring on each day where p is sampled from a uniform $[0, 1]$ distribution. The earliest pickup time (the earliest time a customer can be visited) of a request is uniformly distributed from 9 am to 5 pm on each day. We consider a latest pickup time of 30 minutes after the earliest pick up time. Each request has a latest drop-off time by when the sample has to be delivered to the lab due to medical restrictions; the latest drop-off time for regular requests is 2 hours after its earliest pickup time, and the latest drop-off time for urgent requests is 1 hour after its earliest pickup time (see Table 5.1). We assume all the random requests are urgent requests. We consider that each problem instance has 10 scenarios that are generated by sampling the random requests according to their probability p . We also assume a given number of vehicles to service the requests. The vehicles travel at an average speed of 30 miles per hour to service the requests.

Table 5.1: Time Windows of Regular and Urgent Requests

	Earliest Pickup Time (hrs)	Latest Pickup Time (hrs)	Latest Dropoff Time (hrs)
Regular Request	[9, 17] Uniformly	0.5+Earliest Pickup Time	2 + Earliest Pickup Time
Urgent Request	[9, 17] Uniformly	0.5+Earliest Pickup Time	1 + Earliest Pickup Time

Since the set of regular requests are critical in building the master routes in strategies TAXI and MFIX, in the set of experiments below we consider different proportions of regular to urgent requests. The objective is to identify whether the instance distribution of regular to urgent requests has any bearing on which of the four strategies is more beneficial.

The Tabu search algorithm parameters used in the experiments were selected after a preliminary computational study. This study compared the objective function obtained by using different parameter settings and selected the ones that obtained the best expected objective value. The Tabu search parameters selected for the computational experiments

are $\eta_{max} = 50$, $\gamma_{max} = 50$, $I_{max} = 100$, $\theta_{min} = 10$, and $\theta_{max} = 20$. We note that the results were not significantly sensitive to changes in these parameters and slight variations lead to similar results.

Table 5.2, Table 5.3 and Table 5.4 summarize the computational results with 500 customers, 10 vehicles and different combinations of the cost parameters. The results in these tables correspond to the expected scenario costs, averaged over 10 random instance replications. In these tables, α_t is the unit cost per hour traveled. α_{of} is the fixed cost per trip of taxi usage. α_{ov} is the variable cost per distance the taxi traveled. α_s is the unit cost per count of dissimilarity. Column “Proportion Fix” shows the proportion of deterministic customers among all the potential customers. Column “Strategy” lists the four compared strategies. Column “Travel” shows the total time that a vehicle travels per day on average. Column “Taxi Cost” shows the average daily taxi cost. Column “Dissimilarity” shows the average dissimilarity, which is the total number of vehicles used in the daily routes that is different than the one in the master routes. If a taxi is used, then the dissimilarity is increased by one, as we assume that a different taxi is used each time one is needed. Column “#Taxi Trips” shows the total number of daily taxi trips introduced on average. Column “Travel/Request” shows the average time that a vehicle travels to service a request on a daily basis. Column “Total Cost” shows the average daily total cost including travel cost, taxi cost, and cost on dissimilarity. It is the summation of each type of costs weighted by the unit cost of that type. It does not correspond to the objective function of the model presented in section 3 as it does not include the travel cost of the master route.

Table 5.2: Results with High Taxi & Low Dissimilarity Weights

#Customers: 500; #Vehicles: 10; $\alpha_t=1$, $\alpha_{of}=100$, $\alpha_{ov}=0.5$, $\alpha_s=0.01$;							
Proportion Fixed	Strategy	Travel	Taxi Cost	Dissimilarity	# Taxi Trips	Travel/Request	Total Cost
0.8	TAXI	6.35	5724.57	57.17	57.17	0.16	5788.62
	IND	7.11	968.58	347.14	9.67	0.16	1043.20
	MFIX	7.00	1210.95	72.18	12.09	0.16	1281.69
	MHALF	7.14	1047.74	52.76	10.46	0.16	1119.64
0.6	TAXI	5.13	10132.15	101.19	101.19	0.17	10184.44
	IND	7.03	775.28	327.38	7.74	0.18	848.89
	MFIX	6.86	713.18	112.56	7.12	0.17	782.92
	MHALF	7.05	870.47	59.88	8.69	0.18	941.55
0.4	TAXI	3.70	14873.09	148.54	148.54	0.18	14911.57
	IND	6.81	499.83	300.63	4.99	0.19	570.91
	MFIX	6.53	327.55	162.14	3.27	0.19	394.43
	MHALF	6.72	456.81	68.69	4.56	0.19	524.69
0.2	TAXI	2.22	19846.46	198.21	198.21	0.22	19870.60
	IND	6.40	182.31	271.85	1.82	0.22	248.99
	MFIX	6.32	218.37	208.59	2.18	0.21	283.67
	MHALF	6.22	219.40	78.57	2.19	0.21	282.40

Table 5.3: Results with High Taxi & High Dissimilarity Weights

#Customers: 500; #Vehicles: 10; $\alpha_t=1$, $\alpha_{of}=100$, $\alpha_{ov}=0.5$, $\alpha_s=100$;							
Proportion Fixed	Strategy	Travel	Taxi Cost	Dissimilarity	# Taxi Trips	Travel/Request	Total Cost
0.8	TAXI	6.41	5724.57	57.17	57.17	0.16	11505.69
	IND	7.14	1138.83	354.36	11.37	0.16	36646.26
	MFIX	7.03	1384.21	57.17	13.82	0.16	7171.46
	MHALF	7.24	1371.22	35.98	13.69	0.17	5041.67
0.6	TAXI	5.25	10132.15	101.19	101.19	0.17	20303.69
	IND	6.98	701.17	328.71	7.00	0.18	33641.96
	MFIX	6.90	828.33	101.19	8.27	0.18	11016.37
	MHALF	7.19	1099.84	41.48	10.98	0.18	5319.76
0.4	TAXI	3.79	14873.09	148.54	148.54	0.18	29765.04
	IND	6.72	454.77	302.92	4.54	0.19	30814.00
	MFIX	6.69	437.73	148.54	4.37	0.19	15358.64
	MHALF	6.99	796.36	47.27	7.95	0.20	5593.22
0.2	TAXI	2.25	19846.46	198.21	198.21	0.23	39689.96
	IND	6.38	224.38	272.34	2.24	0.22	27522.17
	MFIX	6.46	239.41	198.21	2.39	0.22	20125.01
	MHALF	6.52	359.63	53.54	3.59	0.22	5778.80

Table 5.4: Results with Low Taxi & High Dissimilarity Weights

#Customers: 500; #Vehicles: 10; $\alpha_t=1$, $\alpha_{of}=0.5$, $\alpha_{ov}=0.5$, $\alpha_s=100$;							
Proportion Fixed	Strategy	Travel	Taxi Cost	Dissimilarity	# Taxi Trips	Travel/Request	Total Cost
0.8	TAXI	6.26	36.16	57.17	57.17	0.16	5815.79
	IND	6.73	12.99	353.80	19.03	0.16	35460.30
	MFIX	6.87	9.45	57.17	14.19	0.16	5795.20
	MHALF	7.28	9.17	35.98	13.88	0.17	3679.95
0.6	TAXI	5.14	63.74	101.19	101.19	0.17	10234.18
	IND	6.46	12.53	328.73	18.27	0.17	32950.16
	MFIX	6.76	8.77	101.19	13.03	0.17	10195.34
	MHALF	7.16	7.91	41.48	11.88	0.18	4227.55
0.4	TAXI	3.73	93.37	148.54	148.54	0.18	14984.68
	IND	6.15	11.64	301.84	16.97	0.18	30257.15
	MFIX	6.33	8.88	148.54	13.03	0.19	14926.16
	MHALF	6.92	5.71	47.27	8.50	0.20	4801.94
0.2	TAXI	2.21	124.57	198.21	198.21	0.22	19967.66
	IND	5.65	11.87	271.81	17.30	0.20	27249.35
	MFIX	5.82	10.35	198.21	15.09	0.21	19889.57
	MHALF	6.39	3.19	53.54	4.67	0.22	5421.10

From the computational results, we make the following observations:

- I. Strategy TAXI has the smallest travel time and largest taxi cost, because of its inability to use the slack times to accommodate the random requests with the regular fleet. Strategies MFIX and MHALF have similar “travel time per request” as strategy IND, which does not aim to coordinate solutions across scenarios.
- II. Strategy IND, which does not use a master route, has the largest dissimilarity; strategy MHALF has the lowest dissimilarity. The process of forming master routes with the deterministic requests and a number of random requests of high probability of occurrence creates daily routes that are similar from day to day, without sacrificing much in routing efficiency.
- III. When the unit cost for route dissimilarity increases (from 0.01 to 100), the dissimilarity for strategies with master routes decreases and the routing costs (travel time and taxi cost) increases. This is because when we give a higher weight on dissimilarity, the routing solution favors reducing dissimilarity at the expense of increasing travel time and taxi cost. The change in the unit cost for route dissimilarity does not significantly impact the solutions of strategies TAXI and IND. This is because in strategy TAXI, the dissimilarity is due to the random requests handled by taxi, which remain the same for any set of parameters; for strategy IND, as there is no master route the dissimilarity is measured against the master route from strategy MFIX, which also is not sensitive to changes in the unit cost for dissimilarity.
- IV. When the fixed unit taxi cost decreases (from 100 to 0.5), while all the other parameters remain the same, there is more taxi use represented by the number of taxi trips. This implies that as taxi usage becomes inexpensive, it becomes a more economical solution to use taxi rather than rerouting to pick up packages by the regular fleet of vehicles.

5.2 Results with Actual Data

We tested our routing approach also using real-life data collected from a leading healthcare provider in Southern California. There are two types of requests in the data set. One is regular daily requests, which needs to be visited every day at a specific time. The

other is random requests that are currently being outsourced to a taxi service. We compared three strategies with this set of data.

- 1) MD Routes: Include a customer into the master route if the pickup and delivery location of a request has a probability of occurring higher than a threshold. In this case, we use a threshold of 10% in order to have enough clients to build a master route since in the actual data set few clients appear frequently. Recourse is used on a daily plans.
- 2) Industry Reroute: Take the existing master route from the healthcare provider as the simulated master routes. Recourse for daily plans.
- 3) Industry Taxi: Take the existing master route from the healthcare provider as the daily routes. Use Taxi for all the random requests.

In the above strategies, the recourse action means dropping the non-occurring requests and inserting the occurring requests on a daily basis. It should be noted that Industry Taxi is the current practice of this healthcare provider. In the following experiments with 30 scenarios, there are 85 deterministic requests and 100 potential random requests on each day and the occurrence probability on each day for these random requests vary from 0 to 0.20. These requests are scattered across 16 medical centers and the pickup of the requests can be at any of these centers with the deliveries being at the central lab (depot). The time windows are 4 hours for regular requests and 2 hours for urgent requests. On a daily basis, 14 regular fleet vehicles are available to service the requests.

The computational results are shown in

Table 5.5, and we see that strategy “Industry Taxi” has the shortest average travel time. The table also shows that the taxi cost and the number of taxi trips of strategy “Industry Taxi” are significantly higher than those of the strategies with recourse actions (MD Routes and Industry Reroute). This implies that, with the recourse technique, we are able to better utilize the slack time on the vehicles to reduce the taxi cost. Meanwhile, even though the average total travel time of a vehicle is higher with the strategies with recourse actions, the average travel time spent for each customer request is lower with these strategies. From the table, we also see that the proposed strategy – MD Routes has the smallest taxi cost and average travel time per request. This shows that the proposed

strategy not only better utilizes the slack time to reduce the taxi cost, but is also an efficient routing solution with the least travel time spent on each request.

Besides the reduction in taxi cost, MD Routes significantly reduced the route dissimilarity. In general, any strategy with a rerouting technique has smaller dissimilarity as the location visited in the master route is going to be more frequently repeated in the daily plans. And the strategy we propose is the best in generating similar routes. This is achieved by having the proposed strategy “MD Routes” including the high probabilistic customers into the master routes; whereas the strategy “Industry Reroute” has only the deterministic customers in the master route.

The results of the analysis with real-life data shows that our heuristic can improve the routing solution by decreasing the taxi and dissimilarity costs. With the current available vehicles and deterministic requests, and sampling on current data set, our heuristic beats the current industry solution by reducing the taxi cost by 45%-48% and reducing dissimilarity by 26%-33%. If we compare with the daily routes obtained by applying the recourse actions on a master route taken from the current industry practice, our heuristic reduces the taxi cost by 16%-17% and it reduces dissimilarity by 9%-12%.

Table 5.5: Results with Actual Data

$\alpha_t=1, \alpha_{of}=100, \alpha_{ov}=0.5, \alpha_s=0.01;$						
Strategy	Travel (hours/day)	Taxi Cost (\$/day)	Dissimilarity# (counts/day)	Taxi Trips (trips/day)	Travel/Request (hours/day)	Total Cost (\$/day)
MD Routes	8.78	5221.10	148.00	52.10	0.03	5345.50
Industry Reroute	8.36	6223.40	164.00	62.10	0.03	6342.10
Industry Taxi	7.24	10023.80	200.00	100.00	0.04	10127.20
$\alpha_t=1, \alpha_{of}=100, \alpha_{ov}=0.5, \alpha_s=100;$						
Strategy	Travel (hours/day)	Taxi Cost (\$/day)	Dissimilarity# (counts/day)	Taxi Trips (trips/day)	Travel/Request (hours/day)	Total Cost (\$/day)
MD Routes	8.82	5291.13	143.40	52.77	0.03	19754.60
Industry Reroute	8.47	6333.60	157.23	63.17	0.03	22175.51
Industry Taxi	7.24	10023.77	200.00	100.00	0.04	30125.17
$\alpha_t=1, \alpha_{of}=0.5, \alpha_{ov}=0.5, \alpha_s=100;$						
Strategy	Travel (hours/day)	Taxi Cost (\$/day)	Dissimilarity# (counts/day)	Taxi Trips (trips/day)	Travel/Request (hours/day)	Total Cost (\$/day)
MD Routes	8.13	43.60	134.23	54.57	0.03	13580.76
Industry Reroute	8.08	49.79	152.93	64.20	0.03	15456.24
Industry Taxi	7.24	73.77	200.00	100.00	0.04	20175.17

6. Conclusions

In this study, we consider a Courier Delivery Problem (CDP), a variant of the Multi-trip Vehicle Routing Problem (MVRP) with uncertainty in customer occurrence and urgency in customer demands. We present a mixed integer programming problem formulation for an example application of the transportation of medical specimens. We develop a heuristic to solve this problem based on insertion and tabu search. Our model represents the probabilistic nature of customer occurrence using scenario-based stochastic programming with recourse. We benefit from the simplicity and flexibility of a master route with daily recourse actions.

Our model first includes a master route problem which represents the uncertainty in the customer occurrence by the probabilities customers are likely to appear and addresses the urgency in delivery time windows by use of the fleet of vehicles in multiple trips. We then define a recourse action of partial rescheduling of routes by omitting non-occurring customers and rescheduling new customers. The master routes created consider efficiency in routing, to represent slack time for accommodating random requests. The daily plans created take into account the efficiency in routing, efficiency in alternative third party courier, as well as route similarities to boost the quality of service. To solve large size problems of the model, we develop a heuristic based on insertion and tabu search.

We explore experimentally the sensitivity of our heuristic on randomly generated problems and a real problem from industry. Experiments on randomly generated problems include sensitivity analysis in varying problem size, customer uncertainty scenarios, resource availability and cost parameters. We compare the quality of the solution with independent daily scheduling, and the industry standard solution. In the experiments on the real problem, we compare the quality of the heuristic solution with the current industry practice. Sensitivity analysis on varying cost parameters shows that our heuristic produces a better solution than the current industry practice by significantly reducing the cost of taxi use and improving route similarity.

References

Alonso, F., Alvarez, M. J., and Beasley, J. E. (2008). A tabu search algorithm for the periodic vehicle routing problem with multiple vehicle trips and accessibility restrictions. *Journal of the Operational Research Society* **59**: 963-976.

Angelelli, E. and Speranza, M.G. (2002). The periodic vehicle routing problem with intermediate facilities. *European Journal of Operational Research* **137**: 233-247.

Astion, M. L., Shojania, K. G., Hamill, T. R., Kim, S., and Ng, V. L. (2003). Classifying laboratory incident reports to identify problems that jeopardize patient safety. *American Journal of Clinical Pathology* **120**: 18-26.

Azi, N., Gendreau, M., and Potvin, J. (2006). An exact algorithm for a single-vehicle routing problem with time windows and multiple routes. *European Journal of Operational Research* **178**: 755–766.

Barrera, J., Homem-de-Mello, T., Moreno, E., Pagnoncelli, B.K., Canessa, G. (1990). Chance-constrained problems and rare events: an importance sampling approach. Working Paper Universidad Adolfo Ibañez.

Bertsimas, D. (1992). A vehicle routing problem with stochastic demand. *Operations Research* **40**: 574-585.

Bertsimas, D. and Simchi-Levi, D. (1996). A new generation of vehicle routing research: robust algorithms, addressing uncertainty. *Operations Research* **44**(2): 286–304.

Brandao, J. and Mercer, A. (1997). A tabu search algorithm for the multi-trip vehicle routing and scheduling problem. *European Journal of Operational Research* **100**: 180-191.

Brandao, J. and Mercer, A. (1998). The multi-trip vehicle routing problem. *Journal of the Operational Research Society* **49**: 799-805.

Campbell, A. M. and Savelsbergh, S. (2004). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science* **38**(3): 369-378.

Francis, P. and Smilowitz, K. (2006). Modeling techniques for periodic vehicle routing problems. *Transportation Research Part B: Methodological* **40**: 872-884.

Gendreau, M., Laporte, G., and Seguin, R. (1995). An exact algorithm for the vehicle routing problem with stochastic demands and customers. *Transportation Science* **29**: 143-155.

Gendreau, M., Laporte, G., and Seguin, R. (1996). Stochastic vehicle routing. *European Journal of Operational Research* **88**: 3-12.

Groër, C., Golden, B., and Wasil, E. (2009). The consistent vehicle routing problem. *Manufacturing and Service Operations Management* **11**: 630-643.

Haughton, M. (2000). Quantifying the benefits of route reoptimization under stochastic customer demands. *Journal of the Operational Research Society* **51**: 320–322.

Haughton, M. and Stenger, A. (1998). Modeling the customer service performance of fixed routes delivery systems under stochastic demands. *Journal of Business Logistics* **9**: 155–172.

Hemmelmayr, V. C., Doerner, K. F., and Hartl, R. F. (2009). A variable neighborhood search heuristic for periodic routing problems. *European Journal of Operational Research* **195**: 791-802.

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

Kellerer, H., Pferschy, U., and Pisinger, D. (2004). *Knapsack Problems*, SpringerVerlag: Berlin Heidelberg.

Laporte, G., Louveaux, F.V., and Mercure, H. (1994). A priori optimization of the probabilistic traveling salesman problem. *Operations Research* **42**: 543-549.

Lin, S. (1965). Computer solutions of the traveling salesman problem. *Bell Systems Technical Journal* **44**: 2245-2269.

Martínez, L. and Amaya, C-A (2013). A vehicle routing problem with multi-trips and time windows for circular items. *Journal of the Operational Research Society* **64**: 1630-1643.

Osman, I. H. (1993). Metastrategy simulated annealing and tabu search algorithms for the vehicle routing problem. *Annals of Operations Research* **41**: 421-451.

Petch, R. J. and Salhi, S. (2003). A multi-phase constructive heuristic for the vehicle routing problem with multiple trips. *Discrete Applied Mathematics* **133**: 69-92.

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

Ren, Y., Dessouky, M. M., and Ordonez, F. (2010). The multi-shift vehicle routing problem with overtime. *Computers & Operations Research* **37**: 1987-1998.

Salhi, S. and Petch, R. J. (2007). A GA based heuristic for the vehicle routing problem with multiple trips. *Journal of Mathematical Modeling and Algorithms* **6**: 591-613.

Secomandi, N. (2001). A rollout policy for the vehicle routing problem with stochastic demands. *Operations Research* **49**: 796 - 802.

Shortle, J., Rebennack, S., and Glover, F. W. (2014). Transmission-capacity expansion for minimizing blackout probabilities. *IEEE Transactions on Power Systems* **29**: 43-52.

Spivey, M. and Powell, W.B. (2004). The dynamic assignment problem. *Transportation Science* **38**: 399 - 419.

Sungur, I., Ordonez, F., Dessouky, M. M., Ren, Y., and Zhong, H. (2010). A model and algorithm for the courier delivery problem with uncertainty. *Transportation Science* **44**: 193-205.

Taillard, E., Laporte, G., and Gendreau, M. (1996). Vehicle routing with multiple use of vehicles. *Journal of the Operational Research Society* **47**: 1065–1070.

Tarantilis, C. D., Stavropoulou, F., and Repoussis, P. P. (2012). A template-based tabu search algorithm for the consistent vehicle routing problem. *Expert Systems with Applications* **39**: 4233–4239..

Yang, W. H., Mathur, K., and Ballou, R. H. (2000). Stochastic vehicle routing problems with restocking. *Transportation Science*, **34**: 99-112.

Zapfel, G. and Bogla, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics* **113**: 980-996.

Zhong, H., Hall, R. W., and Dessouky, M. M. (2007). Territory planning and vehicle dispatching. *Transportation Science* **41**: 74-89.