Routing Courier Delivery Services with Urgent Demand

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

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

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

*corresponding author
Routing Courier Delivery Services with Urgent Demand

Abstract

Courier delivery services deal with the problem of routing a fleet of vehicles from a depot to service a set of customers that are geographically dispersed. In many cases, in addition to a regular routine demand, the industry is faced with sporadic, tightly constrained, urgent requests. 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 the proposed model. We evaluate the model and solution strategy through simulations on randomly generated data as well as on a real data set provided by a leading healthcare provider in Southern California. We find that our solution obtains significant improvement in travel costs as well as in quality of service over alternative methods.

Keywords:
Stochastic vehicle routing; Multi-trip; Time windows; Urgent demand; Insertion; Tabu Search
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 (overnight delivery overseas or 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-period 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 assumption is that a routing solution with additional capacity, which can be used to 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, approaches such as robust optimization or chance constraint models when there are rare urgent requests, will either ignore the unlikely requests or lead to a high cost solution. Instead, in this work we use a stochastic programming with recourse approach to handle the urgent requests. This approach requires a massive number of scenarios, leading to a large scale routing problem. We propose a two stage stochastic programming model to take into account the cost of routing urgent requests with that of routing regular ones. 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 daily 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. It would therefore be preferable, in terms of quality of service, a first stage solution that does not change too much on each scenario due to the recourse action. In this paper we propose a model and solution algorithm for this 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 multiple 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. The
The proposed heuristic solution algorithm builds a first stage solution, or master plan, that will require little modification when adapted for each uncertainty scenario by limiting the recourse actions considered. This model and algorithm are designed based on the assumption that it may be possible to satisfy regular demand with enough slack that some urgent requests can be accommodated so that it is efficient overall.

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. In section 5, experimental results of an application of the proposed heuristic on randomly generated data sets as well as on a real data set from a large healthcare provider in Southern California are presented and discussed. Conclusions are presented 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) improved the study by considering not multi-trip VRP, but 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 outsources 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.

Another variant of the multi-trip VRP is the periodic VRP, which customers have to be visited once or several times in the planning horizon (Angelelli and Speranza, 2002). PVRP extends the classic planning horizon to 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 demand (VRPSD) (e.g., Yang et al., 2000), the vehicle routing with stochastic customer (VRPSC) (e.g., Waters 1989), the vehicle routing problem with stochastic customer and demand (VRPSCD) (e.g., Gendreau et al., 1995), and the probabilistic travelling salesman problem (PTSP) (e.g., 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 (e.g., 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 helps 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).

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 plan 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, taking into account the efficient scheduling of regular and urgent requests, as well as route similarities. The primary distinction in the domain of the multi-trip VRP formulation is that the earlier research on MVRP has discrete operation periods of equal length for all vehicles, and we allow in this work continuous non-equal operation periods for different vehicles. 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 of a week, where the length of a trip of a vehicle is 8 hours per day. In our problem, the vehicles operate in multiple trips during the planning horizon; the length of the trips for each vehicle will not be defined initially, but be flexibly based on the time window of the demands. There are an a-priori undetermined number of trips with varying length during the planning horizon. The numbers of trips will be determined by the demand, since when a vehicle picks up a medical specimen it has to be delivered to the lab within its time window, which might be small if the delivery request is urgent. In this section, we provide a mixed integer 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 delivery 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 $\{1, \ldots, \delta\}$, with scenario $d$ occurring with a given probability $p_d$. We consider a scenario $d=0$ to represent a nominal or expected scenario. There is one depot (node 0) located at the central lab. Each vehicle should leave the depot at the beginning of the day, and return to the depot at the end of the day. It can also return to the 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 lab to keep track of which trip the request is on. The notation of the model formulation is as follows.

The routing parameters:
\( D \): set of scenarios \( D = \{0, 1, ..., \delta \} \).
\( C \): set of customers, \( C = \{1, ..., n\} \).
\( K \): set of vehicles.
\( W \): set of daily trips of a vehicle, \( W = \{1, ..., n\} \).

The cost parameters:
\( t_{ij} \): travel time between node \( i \) and \( j \).
\( \alpha_k \): unit travel cost, dollars per unit time.
\( \alpha_o \): unit outsource cost, dollars per taxi trip.
\( \alpha_s \): unit dissimilarity cost, dollars for each count of dissimilarity.

The stochastic parameters:
\( \rho_{d} \): 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:
\( x_{ijk}^{d} = \begin{cases} 1, & \text{if vehicle } k \text{ travels from node } i \text{ to } j \text{ on scenario } d \\ 0, & \text{otherwise} \end{cases} \)
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 $d = 0$ models the expected situation. We refer to the solution in this scenario as the master route and will be the routing solution with respect to which the similarity of routes on scenario $d$ 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 to the lab and back to the depot without spending any actual time.

3) $t_{ij}$ is the travel time between node $i$ and $j$. Particularly, $t_{0i}$ is the travel time between the depot and node $i$; $t_{i(n+1)}$ is the travel time between node $i$ and the depot.

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 if customer $i$ is visited by

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

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

$y_{ik}^{d}$: 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 $d$.
vehicle \( k \) either on scenario \( d \) or 0, but not both. This variable \( r_{ik}^{d} \) equals to 0 if customer \( i \) is visited by vehicle \( k \) both on scenario \( d \) and 0, or is not visited in either scenario. In other words the dissimilarity is counted as one if a customer is visited by a different vehicle than in the master plan.

Problem formulation:

Minimize

\[
\alpha_t \cdot \sum_{d \in D} \sum_{k \in K} \left( \sum_{i \in C^d} \sum_{j \neq i} t_{ij} \cdot x_{ijk}^d + \sum_{w \in W} \sum_{i \in C^d} t_{0i} \cdot x_{0ik}^w + \sum_{i \in D} t_{i(n+1)}(n+1) \cdot x_{i(n+1)k}^w \right) \\
+ \alpha_0 \cdot \sum_{d \in D} \sum_{i \in C^d} u_i^d + \alpha_s \cdot \sum_{d \in D \setminus \{0\}} \sum_{k \in K} \sum_{i \in C^d} r_{ik}^d
\]

(3.1)

Subject to:

Routing constraints:

\[
\sum_{k \in K} \sum_{i \in C^d} x_{ijk}^d = \sum_{i \in C^d} x_{i(n+1)k}^d + u_i^d = 1 \quad i \in C^d, d \in D \tag{3.2}
\]

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

\[
\sum_{w \in W} x_{0wk}^w \leq \sum_{i \in C^d} x_{i(n+1)k}^d \quad k \in K, w \in W, d \in D \tag{3.4}
\]

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

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

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

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

\[
y_{i(n+1)k}^w \leq y_{i(n+1)k}^{(n+1)d} \quad w \in W, d \in D, k \in K \tag{3.9}
\]

\[
a_i^d \leq y_{ik}^d \leq b_i^d \quad i \in C^d, k \in K, d \in D \tag{3.10}
\]
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. For brevity, we will use the term “taxi” to refer to an outsourced vehicle for the remainder of the paper. The travel cost is represented by \( \sum \sum \sum \sum \sum \sum \sum \), 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 \( \alpha_o \cdot \sum_d \rho_d \sum_{i \in C_d} u_i^d \), which is proportional to the
expected number of trips that a taxi is used to handle the demands unmet by the regular fleet. It should be noted that this term could easily include the total taxi distance if we change it to \( \alpha_o \cdot \sum_{d \in D} p_d \sum_{i \in E} u_{d,i} t_{oi} \). The expected route dissimilarity cost is measured by \( \alpha_s \cdot \sum_{d \in D \setminus \emptyset} p_d \sum_{k \in K} \sum_{i \in E} r_{d,ik} \), which is proportional to the total number of customers that are serviced by a vehicle different from the one servicing it in the master plan.

There are two groups of constraints in our model, namely routing constraints and domain constraints. Constraint (3.2) assures on each day that each customer should be visited directly from the depot, right after a vehicle services customer \( j \), or by a taxi when the regular fleet is unavailable. Constraint (3.3) assures that each vehicle must leave the customer after visiting it. It also addresses the fact that a customer has to be visited by a vehicle in one of its trips in a day. Constraint (3.4) ensures that each individual trip should start with leaving the depot and end by returning to the depot. Constraint (3.5) enforces 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. Constraint (3.6) assures the relationship of arrival times at customers \( i \) and \( j \), when customer \( j \) is visited right after \( i \) is visited. Constraint (3.7) expresses the relationship of arrival time to customer \( j \), when \( j \) is the first customer request a vehicle handles in a trip. Constraint (3.8) expresses the relationship of arrival time to customer \( j \), when \( j \) is the last customer request a vehicle handles in a trip. Constraint (3.9) enforces 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. These constraints, which correspond to an adaptation of MTZ constraints to a multi-trip model, eliminate infeasible subtours. Constraint (3.10) enforces the arrival time of a vehicle at a customer to be in the required time window for handling the customer request. Constraint (3.11) requires that the arrival time at a customer on a trip should be between the start time and the end time of the trip. Constraint (3.12) requires that each vehicle should visit the lab before the drop-off deadline of each specimen collected by a vehicle on a trip. Constraint (3.13) is another representation of our expression for dissimilarity \( r_{d,ik} = |\sum_{w \in W} z_{ik}^{wd} - \sum_{w \in W} z_{ik}^{wd}| \). It removes 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.
4. Heuristic

Because of the combinatorial nature of the problem, exact solution methods will only be able to solve small size instances of this problem. Since there are $|D|$ scenarios that have to be taken into account in the routing, and each vehicle makes $n$ trips 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 $nk$ vehicles. 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 420 vehicles on 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 plans, and constructing daily plans. The central idea of the heuristic is to separate the problem for each day $d \in D$ 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 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 day $d$. The insertion and tabu search procedures are used to construct efficient routes in both parts of the heuristic (master and daily routes).

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 day and start 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 or taxi cost.

for all the positions in all the activated routes
    Find the feasible insertion positions with minimum insertion cost;
if the insertion is feasible then
    Update the routes;
elseif there is a vehicle to activate then
    Put the request on the new vehicle;
else update the taxi cost;

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 day, vehicles and trip for simplicity, we can express the arrival time to node $i$ as follows: $y_i = \max(y_{i-1}, a_{i-1}) + t_{i-1,i}$, 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 $i$.

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.

for all the positions in all routes
    find the feasible insertion positions with minimum insertion cost;
calculate taxi cost;
if minimum insertion cost is smaller than taxi cost
then use fleet;
else use taxi;
if use fleet
then update the routes;
if use taxi or infeasible to insert
then update the taxi cost;

The feasibility of an insertion is then verified if both the specimen pickup and its drop off at the lab are within its 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 in proportion 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

<table>
<thead>
<tr>
<th>Pickup</th>
<th>Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-2</td>
<td>i-1</td>
</tr>
<tr>
<td>i</td>
<td>i+1</td>
</tr>
</tbody>
</table>
Figure 4.2: Delivery occurs a+1 stops after pickup.

<table>
<thead>
<tr>
<th>Pickup</th>
<th>Delivery</th>
</tr>
</thead>
<tbody>
<tr>
<td>i-2</td>
<td>i+a-1</td>
</tr>
<tr>
<td>i-1</td>
<td>i+a</td>
</tr>
<tr>
<td>i</td>
<td>i+a+1</td>
</tr>
</tbody>
</table>

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 less 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 the standard 2-opt exchange move (Lin, 1965) and the 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 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) on a 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 plan or a daily plan to improve  
**Output:** improved master plan or daily plan

```
repeat
```
randomly chose two routes from the solution
generate $\eta_{max}$ neighbors from $\lambda$-interchange operator
generate $\gamma_{max}$ neighbors from 2-opt operator
choose the best solution and make the move;
randomly generate a tabu tenure $\theta$ from a uniform distribution $U(\theta_{min}, \theta_{max})$;
if the move is $\lambda$-interchange then
    set the tabu for moving the exchanged requests for $\theta$ iterations;
else
    set the tabu for moving the exchanged nodes for $\theta$ iterations;
until no improvement in $I_{max}$ iterations;
calculate the objective and save the current solution;

In each iteration, the Tabu search generates $\eta_{max}$ $\lambda$-interchange neighbors and $\gamma_{max}$ 2-opt neighbors of the current solution. The number of Tabu iterations $\theta$ 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 overwritten 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 throughout the day. A customer that requests service every day usually has wide time windows and should be considered a regular request in the master plan. 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 each day, thus incurring a small additional cost to adapt to the actual customers that appear on day $d$. 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 Plan

**Input:** All the customers to insert; the probability of a customer to request service in a day; 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 plan by calling Algorithm 1.1;
    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 plan, 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 plan 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 day.

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

<table>
<thead>
<tr>
<th>Algorithm 4: Formation of Daily Plans</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input: the master plan; daily requests</td>
</tr>
<tr>
<td>Output: the daily plans</td>
</tr>
<tr>
<td>for each day do</td>
</tr>
<tr>
<td>take the master plan (generated by Algorithm 3) as the initial daily plan;</td>
</tr>
<tr>
<td>for all the requests in the master plan</td>
</tr>
<tr>
<td>if the request does not occur on the day then</td>
</tr>
<tr>
<td>drop the request from the daily plan;</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>for all the requests on the day do</td>
</tr>
<tr>
<td>if a request is NOT included in the master plan then</td>
</tr>
<tr>
<td>insert the request into the daily plan by calling Algorithm 1.1;</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>improve the daily plan with Tabu search by calling Algorithm 2;</td>
</tr>
<tr>
<td>for all the requests serviced by taxi do</td>
</tr>
<tr>
<td>try inserting the request into the daily plan again by calling Algorithm 1.1;</td>
</tr>
<tr>
<td>end for</td>
</tr>
<tr>
<td>end for</td>
</tr>
</tbody>
</table>
5. Experimental Results

5.1 Results on Randomly Generated Data Sets

We first test our heuristic using simulation on randomly generated data sets. Consider a city to be serviced with a two-dimensional coordinate system. The boundary of the city 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 the potential customers are known a priori, and the potential customers, in each experiment, are uniformly distributed in the city. Some customers request service at a fixed time every day (regular deterministic requests), while others only request service at a fixed time on some of the days (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. The latest pickup time (the latest time a customer can be visited) of the request is 30 minutes after the corresponding earliest pickup time. Each request has a latest drop-off time (a deadline by which the sample has to be delivered to the lab); 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 also assume a given number of vehicles to service the requests, which might be different in each experiment. The vehicles travel at an average speed of 30 miles per hour to service the requests.

<table>
<thead>
<tr>
<th></th>
<th>Earliest Pickup Time (hours)</th>
<th>Latest Pickup Time (hours)</th>
<th>Latest Drop-off Time (hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular Request</td>
<td>[9, 17] Uniformly</td>
<td>0.5 + Earliest Pickup Time</td>
<td>2 + Earliest Pickup Time</td>
</tr>
<tr>
<td>Urgent Request</td>
<td>[9, 17] Uniformly</td>
<td>0.5 + Earliest Pickup Time</td>
<td>1 + Earliest Pickup Time</td>
</tr>
</tbody>
</table>
We show the simulation results with the above assumptions and data inputs. In each experiment, we assume a fixed number of potential requests, a fixed proportion of deterministic requests among all the requests, and a fixed number of available vehicles to handle the requests. The result of each experiment is taken by averaging the results of 10 replications, each of which takes the average result of 10 scenarios. In each replication, a random request customer is assigned a probability $p$ of occurring, where $p$ is a sample from a uniform $[0, 1]$. In each scenario of a replication, we determine the occurrence of each request by sampling based on the probability $p$.

In each experiment, we compare the following four strategies by 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, on a daily basis.

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 plans 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 plans with a customer occurrence probability threshold of 0.5.)

The parameters we use in the experiments for the Tabu search algorithms are $\eta_{max} = 50$, $\gamma_{max} = 50$, $I_{max} = 100$, $\theta_{min} = 10$, and $\theta_{max} = 20$. Table 5.2, Table 5.3,
and Table 5.4 summarize the simulation results with 500 customers, 10 vehicles and different combinations of the cost parameters. In these tables, \( \alpha_t \) is the unit cost per hour traveled. \( \alpha_{of} \) is the fixed cost per trip of taxi usage. \( \alpha_{ov} \) is the variable cost per hour the taxi traveled. \( \alpha_s \) 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. Moreover, as there is no master routes generated for independent scheduling, the dissimilarity is calculated by comparing the daily routes to the master routes generated in strategy “master fix”. Column “#Taxi Trips” shows the total number of daily taxi trips introduced on average. Column “Travel/Request” shows the time that a vehicle travels to service a request on a daily basis on average. 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.
Table 5.2: Simulation Result with High Taxi Cost & Low Similarity Cost

<table>
<thead>
<tr>
<th>Proportion Fixed</th>
<th>Strategy</th>
<th>Travel</th>
<th>Taxi Cost</th>
<th>Dissimilarity</th>
<th># Taxi Trips</th>
<th>Travel/Request</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>TAXI</td>
<td>6.35</td>
<td>5724.57</td>
<td>57.17</td>
<td>57.17</td>
<td>0.16</td>
<td>5788.62</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>7.11</td>
<td>968.58</td>
<td>347.14</td>
<td>9.67</td>
<td>0.16</td>
<td>1043.20</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>7.00</td>
<td>1210.95</td>
<td>72.18</td>
<td>12.09</td>
<td>0.16</td>
<td>1281.69</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>7.14</td>
<td>1047.74</td>
<td>52.76</td>
<td>10.46</td>
<td>0.16</td>
<td>1119.64</td>
</tr>
<tr>
<td>0.6</td>
<td>TAXI</td>
<td>5.13</td>
<td>10132.15</td>
<td>101.19</td>
<td>101.19</td>
<td>0.17</td>
<td>10184.44</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>7.03</td>
<td>775.28</td>
<td>327.38</td>
<td>7.74</td>
<td>0.18</td>
<td>848.89</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.86</td>
<td>713.18</td>
<td>112.56</td>
<td>7.12</td>
<td>0.17</td>
<td>782.92</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>7.05</td>
<td>870.47</td>
<td>59.88</td>
<td>8.69</td>
<td>0.18</td>
<td>941.55</td>
</tr>
<tr>
<td>0.4</td>
<td>TAXI</td>
<td>3.70</td>
<td>14873.09</td>
<td>148.54</td>
<td>148.54</td>
<td>0.18</td>
<td>14911.57</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.81</td>
<td>499.83</td>
<td>300.63</td>
<td>4.99</td>
<td>0.19</td>
<td>570.91</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.53</td>
<td>327.55</td>
<td>162.14</td>
<td>3.27</td>
<td>0.19</td>
<td>394.43</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>6.72</td>
<td>456.81</td>
<td>68.69</td>
<td>4.56</td>
<td>0.19</td>
<td>524.69</td>
</tr>
<tr>
<td>0.2</td>
<td>TAXI</td>
<td>2.22</td>
<td>19846.46</td>
<td>198.21</td>
<td>198.21</td>
<td>0.22</td>
<td>19870.60</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.40</td>
<td>182.31</td>
<td>271.85</td>
<td>1.82</td>
<td>0.22</td>
<td>248.99</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.32</td>
<td>218.37</td>
<td>208.59</td>
<td>2.18</td>
<td>0.21</td>
<td>283.67</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>6.22</td>
<td>219.40</td>
<td>78.57</td>
<td>2.19</td>
<td>0.21</td>
<td>282.40</td>
</tr>
</tbody>
</table>

Table 5.3: Simulation Result with High Taxi Cost & High Similarity Cost

<table>
<thead>
<tr>
<th>Proportion Fixed</th>
<th>Strategy</th>
<th>Travel</th>
<th>Taxi Cost</th>
<th>Dissimilarity</th>
<th># Taxi Trips</th>
<th>Travel/Request</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>TAXI</td>
<td>6.41</td>
<td>5724.57</td>
<td>57.17</td>
<td>57.17</td>
<td>0.16</td>
<td>11505.69</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>7.14</td>
<td>1138.83</td>
<td>354.36</td>
<td>11.37</td>
<td>0.16</td>
<td>36646.26</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>7.03</td>
<td>1384.21</td>
<td>57.17</td>
<td>13.82</td>
<td>0.16</td>
<td>7171.46</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>7.24</td>
<td>1371.22</td>
<td>35.98</td>
<td>13.69</td>
<td>0.17</td>
<td>5041.67</td>
</tr>
<tr>
<td>0.6</td>
<td>TAXI</td>
<td>5.25</td>
<td>10132.15</td>
<td>101.19</td>
<td>101.19</td>
<td>0.17</td>
<td>20303.69</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.98</td>
<td>701.17</td>
<td>328.71</td>
<td>7.00</td>
<td>0.18</td>
<td>33641.96</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.90</td>
<td>828.33</td>
<td>101.19</td>
<td>8.27</td>
<td>0.18</td>
<td>11016.37</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>7.19</td>
<td>1099.84</td>
<td>41.48</td>
<td>10.98</td>
<td>0.18</td>
<td>5319.76</td>
</tr>
<tr>
<td>0.4</td>
<td>TAXI</td>
<td>3.79</td>
<td>14873.09</td>
<td>148.54</td>
<td>148.54</td>
<td>0.18</td>
<td>29765.04</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.72</td>
<td>454.77</td>
<td>302.92</td>
<td>4.54</td>
<td>0.19</td>
<td>30814.00</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.69</td>
<td>437.73</td>
<td>148.54</td>
<td>4.37</td>
<td>0.19</td>
<td>15358.64</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>6.99</td>
<td>796.36</td>
<td>47.27</td>
<td>7.95</td>
<td>0.20</td>
<td>5593.22</td>
</tr>
<tr>
<td>0.2</td>
<td>TAXI</td>
<td>2.25</td>
<td>19846.46</td>
<td>198.21</td>
<td>198.21</td>
<td>0.23</td>
<td>39689.96</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.38</td>
<td>224.38</td>
<td>272.34</td>
<td>2.24</td>
<td>0.22</td>
<td>27522.17</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.46</td>
<td>239.41</td>
<td>198.21</td>
<td>2.39</td>
<td>0.22</td>
<td>20125.01</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>6.52</td>
<td>359.63</td>
<td>53.54</td>
<td>3.59</td>
<td>0.22</td>
<td>5778.80</td>
</tr>
</tbody>
</table>
Table 5.4: Simulation Result with Low Taxi Cost & High Similarity Cost

<table>
<thead>
<tr>
<th>Proportion Fixed</th>
<th>Strategy</th>
<th>Travel Time</th>
<th>Taxi Cost</th>
<th>Dissimilarity</th>
<th># Taxi Trips</th>
<th>Travel/Request</th>
<th>Total Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.8</td>
<td>TAXI</td>
<td>6.26</td>
<td>36.16</td>
<td>57.17</td>
<td>57.17</td>
<td>0.16</td>
<td>5815.79</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.73</td>
<td>12.99</td>
<td>353.80</td>
<td>19.03</td>
<td>0.16</td>
<td>35460.30</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.87</td>
<td>9.45</td>
<td>57.17</td>
<td>14.19</td>
<td>0.16</td>
<td>5795.20</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>7.28</td>
<td>9.17</td>
<td>35.98</td>
<td>13.88</td>
<td>0.17</td>
<td>3679.95</td>
</tr>
<tr>
<td>0.6</td>
<td>TAXI</td>
<td>5.14</td>
<td>63.74</td>
<td>101.19</td>
<td>101.19</td>
<td>0.17</td>
<td>10234.18</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.46</td>
<td>12.53</td>
<td>328.73</td>
<td>18.27</td>
<td>0.17</td>
<td>32950.16</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.76</td>
<td>8.77</td>
<td>101.19</td>
<td>13.03</td>
<td>0.17</td>
<td>10195.34</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>7.16</td>
<td>7.91</td>
<td>41.48</td>
<td>11.88</td>
<td>0.18</td>
<td>4227.55</td>
</tr>
<tr>
<td>0.4</td>
<td>TAXI</td>
<td>3.73</td>
<td>93.37</td>
<td>148.54</td>
<td>148.54</td>
<td>0.18</td>
<td>14984.68</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>6.15</td>
<td>11.64</td>
<td>301.84</td>
<td>16.97</td>
<td>0.18</td>
<td>30257.15</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>6.33</td>
<td>8.88</td>
<td>148.54</td>
<td>13.03</td>
<td>0.19</td>
<td>14926.16</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>6.92</td>
<td>5.71</td>
<td>47.27</td>
<td>8.50</td>
<td>0.20</td>
<td>4801.94</td>
</tr>
<tr>
<td>0.2</td>
<td>TAXI</td>
<td>2.21</td>
<td>124.57</td>
<td>198.21</td>
<td>198.21</td>
<td>0.22</td>
<td>19967.66</td>
</tr>
<tr>
<td></td>
<td>IND</td>
<td>5.65</td>
<td>11.87</td>
<td>271.81</td>
<td>17.30</td>
<td>0.20</td>
<td>27249.35</td>
</tr>
<tr>
<td></td>
<td>MFIX</td>
<td>5.82</td>
<td>10.35</td>
<td>198.21</td>
<td>15.09</td>
<td>0.21</td>
<td>19889.57</td>
</tr>
<tr>
<td></td>
<td>MHALF</td>
<td>6.39</td>
<td>3.19</td>
<td>53.54</td>
<td>4.67</td>
<td>0.22</td>
<td>5421.10</td>
</tr>
</tbody>
</table>

From the simulation 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, thus resulting in servicing fewer requests by the regular fleet. Strategy MFIX and MHALF have similar “travel time per request” as strategy IND, a near-optimal routing solution, suggesting that our master plan strategies provide efficient routing solutions.

II. Strategy IND has the largest dissimilarity; strategy MHALF has the lowest dissimilarity. If we schedule the routing for each day independently without a master plan, the routes become dissimilar from day to day. Even though we get an efficient route as measured in travel time and taxi cost, the quality of service, as measured in route dissimilarity, is poor. If we form master routes with the deterministic requests and a number of random requests of high
probability of occurrence, daily routes are created, which are similar from
day to day, without scarifying much in routing efficiency.

III. When the unit cost for route dissimilarity increases (from 0.01 to 100), the
dissimilarity for strategies with master plans decreases and the routing
efficiency (travel time and taxi cost) increases. This is because when we give
a higher weight on dissimilarity, the routing solution favors less dissimilarity,
and trades that with less routing efficiency. The change in the unit cost for
route dissimilarity does not significantly impact the solutions of strategies
TAXI and IND. The reason is that for strategy TAXI, the dissimilarity is
contributed by the random requests handled by taxi, which remain the same
with any set of parameters; for strategy IND, there is no master plan to use to
construct daily routes, but the dissimilarity is measured against the master
plan from strategy MFIX. Hence, the dissimilarity with IND might even
increase when the unit cost of dissimilarity increases.

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 become 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 plan if the pickup and delivery
location of a request has a probability of occurring higher than a threshold
(e.g., 10%). Recourse for daily plans.

2) Industry Reroute: Take the existing master plan from the healthcare provider
as the simulated master routes. Recourse for daily plans.
3) Industry Taxi: Take the existing master plan 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 simulation 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 and delivery of the requests can be at any of these centers. 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 simulation 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 plan 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 plan.
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 resource of vehicles, the current 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 plan 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: Simulation Results with Actual Data

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Travel (hours/day)</th>
<th>Taxi Cost ($/day)</th>
<th>Dissimilarity (counts/day)</th>
<th># Taxi Trips (trips/day)</th>
<th>Travel/Request (hours/day)</th>
<th>Total Cost ($/day)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MD Routes</td>
<td>8.78</td>
<td>5221.10</td>
<td>148.00</td>
<td>52.10</td>
<td>0.03</td>
<td>5345.50</td>
</tr>
<tr>
<td>Industry Reroute</td>
<td>8.36</td>
<td>6223.40</td>
<td>164.00</td>
<td>62.10</td>
<td>0.03</td>
<td>6342.10</td>
</tr>
<tr>
<td>Industry Taxi</td>
<td>7.24</td>
<td>10023.80</td>
<td>200.00</td>
<td>100.00</td>
<td>0.04</td>
<td>10127.20</td>
</tr>
<tr>
<td>Strategy</td>
<td>8.82</td>
<td>5291.13</td>
<td>143.40</td>
<td>52.77</td>
<td>0.03</td>
<td>19754.60</td>
</tr>
<tr>
<td>MD Routes</td>
<td>8.47</td>
<td>6333.60</td>
<td>157.23</td>
<td>63.17</td>
<td>0.03</td>
<td>22175.51</td>
</tr>
<tr>
<td>Industry Reroute</td>
<td>7.24</td>
<td>10023.77</td>
<td>200.00</td>
<td>100.00</td>
<td>0.04</td>
<td>30125.17</td>
</tr>
<tr>
<td>Strategy</td>
<td>8.13</td>
<td>43.60</td>
<td>134.23</td>
<td>54.57</td>
<td>0.03</td>
<td>13580.76</td>
</tr>
<tr>
<td>MD Routes</td>
<td>8.08</td>
<td>49.79</td>
<td>152.93</td>
<td>64.20</td>
<td>0.03</td>
<td>15456.24</td>
</tr>
<tr>
<td>Industry Reroute</td>
<td>7.24</td>
<td>73.77</td>
<td>200.00</td>
<td>100.00</td>
<td>0.04</td>
<td>20175.17</td>
</tr>
</tbody>
</table>

### 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 an efficient heuristic 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 plan with daily recourse actions.

Our model first includes a master plan 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-life problem collected 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 to an industry standard solution. In the experiments with real-life data, we compare the quality of the solution with the current industry solution with and without recourse action. Sensitivity analysis on varying cost parameters shows that our heuristic produces a better solution than the current practice by significantly reducing the cost on taxi use and improving route similarity.
References


