Territory Planning and Vehicle Dispatching With Driver Learning

Hongsheng Zhong

United Parcel Service
2311 York Road, Timonium MD 21093

Randolph W. Hall
Maged Dessouky

Daniel J. Epstein Department of
Industrial and Systems Engineering
University of Southern California
Los Angeles, CA 90089

December 17, 2004

Abstract

This paper investigates the construction of routes for local delivery of packages. The primary objective of this research is to provide realistic models to optimize vehicle dispatching when customer locations and demands vary from day to day, while maintaining driver familiarity with their service territories, hence dispatch consistency. The objective of increasing driver familiarity tends to make routes or service territories fixed. On the other hand, to serve varying demand, it is advantageous to reassign vehicles/drivers and service territories each day. To balance the tradeoffs between these two objectives, we developed the concepts of "cell", "core area", and "flex-zone", and created a two-stage vehicle routing model -- Strategic Core Area Design and Operational Cell Routing -- and explicitly evaluated the effect of driver familiarity through use of learning and forgetting curves.

1. INTRODUCTION

Over the last 20 years, the parcel distribution industry has experienced a large growth in business. Their next-day and overnight delivery services have helped industries revolutionize manufacturing processes and cut inventory cost through just-in-time techniques, and permitted the rapid delivery of documents around the world. Especially in recent years, with the rapid growth of e-commerce, the demand for courier services from package delivery companies has been strong. In fact, not only has the demand for transporting packages increased, but the demand for integrated services from parcel companies in the fields of logistics, supply chain management and e-commerce has also grown. However, at the same time, competition is becoming more intense. Besides the competition from their own industry, private package carriers are also threatened by USPS (US Postal Service), which has been advertising the low cost and effectiveness of its Priority Mail and parcel post services. Therefore, satisfying customer demand with the least business operation cost has become a major concern. As a result, the need for practical optimization models for daily business operations is increasing in these companies.

This paper investigates the construction of routes for local delivery of packages. The primary objective of this research is to improve the package delivery dispatching operations under uncertainty by building a robust dispatching system, which provides flexible routes, with more familiar drivers – hence providing better service and lower cost. Also the process will be more automated, while allowing dispatchers to participate in the routing process. A significant improvement will be achieved by constructing realistic models to optimize territory planning and vehicle dispatching with random customer locations and demand, while maintaining driver familiarity with their service areas.

Each company in the parcel shipping industry has its own package distribution network. Usually, the network has two distinct parts: longhaul transportation and local distribution. Longhaul transportation deals with freight movement between consolidation terminals, while local distribution deals directly with customers by providing pickup and delivery services within a local area. With a very high volume of packages processed each day in the local distribution center, a good dispatching system that enhances the driver's performance will directly improve the overall service quality and reduce the operating cost. Therefore, efficiently assigning workload and delivery routes to drivers is extremely important to these companies.

Vehicle dispatching addresses the optimization of driver and vehicle assignments to serve customer demand. Vehicle dispatching is usually subject to a number of constraints, such as vehicle capacity and driver work duration. An important objective, which most companies emphasize, is to maximize driver familiarity within their service territories. This familiarity is achieved by assigning the same driver to the same set of customers each day.

Driver familiarity results from visiting service areas repeatedly. With increased familiarity, driver performance increases due to ease in finding addresses and locations within buildings, and efficiency in organizing routes. At the same time, it is also very important for a dispatcher to have flexibility to optimize dispatch plans on a daily basis by adjusting the number of vehicle routes according to changing customer locations and demand, so as to maximize the driver utilization and to minimize total routing cost. The objective of increasing driver familiarity tends to make routes or service territories fixed. On the other hand, the objective of increasing flexibility to optimize the number of routes and total distance traveled tends to assign vehicles/drivers to variable routes or service territories each day. Thus, it is very important for a good dispatching system to balance these tradeoffs.

This research originated from a vehicle dispatching problem encountered by United Parcel Service, but the proposed models are applicable to a variety of cases in which local distribution operations are involved (e.g. FedEx, DHL, and many less-than-truckload carriers). The issues studied in this research are important to companies like UPS and FedEx because they need to design a dispatching system that provides consistent dispatch plans while keeping enough flexibility to accommodate the daily variations of customer demand There are many models and methods in the literature that deal with optimizing vehicle dispatching or vehicle routing in a local distribution network. However, it seems that no research has been published related to the value of driver learning or driver familiarity with their service territories, hence trading off the benefits between dispatching consistency and flexibility. The major innovation of this research is developing a methodology for solving the large-scale vehicle dispatching problem under daily variation while accounting for driver familiarity.

2. LITERATURE REVIEW

The problem addressed in this paper is a type of stochastic routing problem in which routes are constructed for multiple days or weeks with random daily variations. Most real-world vehicle routing problems are stochastic in nature; however, it is very difficult to solve stochastic problems, even by heuristic algorithms (Laporte, 1992). Thus, they are often formulated in a way that can finally be reduced to a deterministic case.

In addition to algorithmic research on vehicle routing, approximation models have been studied for predicting the optimal route length and the optimal shape of routing districts. Daganzo (1984) and Newell and Daganzo (1986) created a continuous-space model to represent near-optimal route geometry and approximate the optimal length of routes that are constrained by the number of stops and originate from a single depot. Hall, Du, and Lin (1994) combined the continuous space model and the discrete model into a so-called Integrated Algorithm (IA).

According to the authors, the computational result was good, especially when the problem size is large. The article also showed that Daganzo's continuous space approximation of average route length was accurate for constructed routes. Hall (1996) particularly studied the optimal route designs of pickup and delivery systems for overnight carriers using continuous space approximation models and demonstrated how the constraints of overnight delivery affect the designs.

Deterministic vehicle routing problems (VRP) assume that the set of locations to be visited, and their characteristics (such as shipment size), are know with certainty when routes are constructed. This might occur when a known set of deliveries are routed for an individual day. However, in many instances, the territories served by drivers remain constant, or nearly constant, over a period of time ranging from a week to several months. While the territories remain constant, the set of customers requiring visits and their shipment sizes are likely to vary each day. Thus, it is important to have routing methods that are effective when parameters such as customer demand, travel time, and the customer set are stochastic, leading to Stochastic Vehicle Routing Problems (SVRP). The most commonly researched uncertainty is customer demand. With the randomness inherent in the problem, the objective of the SVRP is usually to minimize the expected total routing cost. SVRP is typically formulated as either Chance-Constrained Programming (CCP) or Stochastic Programming with Recourse (SPR). For a detailed literature on the SVRP, the reader is referred to Gendreau et al (1996).

While most of the literature on SVRP only deals with uncertainty in customer demand, it is not uncommon that the set of customers to be served and their demands are both stochastic in nature (one example is package delivery services). The most commonly researched area is VRP with omitted customers. Waters (1989) proposed three alternatives to deal with omitted customers: fixed route, semi-fixed route and variable route, as well as their potential savings due to different percentages of omitted customers. While Benton and Rossetti (1992) presented a heuristic based on these three alternatives, Haughton (1998) further developed a more accurate model for estimating the expected savings with semi-fixed routes using statistical calibration. Haughton and Stenger (1998) modeled customer service for fixed route delivery systems under stochastic demand. Haughton (2000) developed a framework for quantifying the benefits of route re-optimization under stochastic customer demands. Haughton (2004) considered assignment rules that managers make in order to increase customer driver familiarity. He presented a statistical model of one such rule.

Bertsimas (1992) suggests constructing an "a priori" sequence among all customers instead of re-optimizing the routes when the demand becomes known. Particularly, the author proposed a cyclic heuristic to build a priori sequence in the case that demand becomes known only when the customer is visited. According to the author, this heuristic performs very well from a worst-case perspective, especially if customer demand remains the same.

Recent research that considers the design of large-scale logistic systems for uncertain environments comes from Erera (2000). His dissertation adapts a continuum approximation methodology developed for deterministic problems for analysis of large-scale vehicle dispatching problems. The methodology is to find expected cost approximations that allow near-optimal configuration of such systems under stochastic customer locations and demand. Two fundamental logistics systems were studied: load-constrained vehicle routing with uncertain customer locations and demand, and deadline vehicle routing with uncertain customer service times. This research adapts a coordinated design, which allows neighboring vehicles or the whole fleet to pool capacity. They are called locally-coordinated design and globally-coordinated design. We will see in later sections that our research provides similar ideas, which we call flex-zones, to coordinate the fleet both locally and globally. Our method has advantages in that customers that are assigned to core drivers are served by the same driver each day, which guarantees a high level-of-service for a large portion of customers, whereas in Erera's research the customers that are a priori assigned to driver territories still can be served by other drivers.

It is also possible for customer locations and demand to be revealed at the beginning of a day. A simple solution for this kind of period routing is to repeat the routing solution procedure each day based on that day's requirements; this is called variable routes. But for various reasons, such as tradeoffs on administrative inconvenience and driver unfamiliarity with customers, complete variability is undesirable. Alternately, a fixed route problem constructs routes that remain constant over many days and in the implementation, certain recourse procedures need to be adapted to deal with route failure. Both stop sequence and service territory can be fixed in this problem. Christofides (1971) was the first to investigate the fixed routes problem. He proposed a solution approach that first solves the daily routing problem based on typical daily customer demand data and then forms the fixed routes by adding inter-customer links according to their frequency of occurrence. Beasley (1984) adapted the savings algorithm from Clarke and Wright (1964) and a k-exchange procedure by Lin (1966) to the fixed routes problem. The computational result according to the author indicated that for a demand variation of ± 10%

the increase in travel distance is less than 2% compared to deterministic routing for each instance. Wong and Beasley (1984) generalized the fixed routes problem by developing fixed delivery areas instead of fixed routes.

Savelsbergh and Goetschalckx (1995) studied the viability of fixed routes as an alternative to variable routes. They proposed a two-phase method. In the first phase, a generalized assignment heuristic constructs an initial solution based on mean customer demand. In the second phase, a local search procedure improves the current solution by minimizing the objective function of a stochastic program with recourse. According to the authors, the lengths of the routes generated by this algorithm were within 10% for coefficients of variations smaller than 0.3 of the customer demand. If the number of customers on a route is larger than six, the length penalty decreased further to 5%. They argue that this indicates that fixed routes do provide a viable alternative to daily recomputed routes.

Beasley and Christofides (1997) studied the delivery operation of a large mail order/catalogue company, which prefers a set of fixed delivery routes in each day. By aggregating the customers according to postcode, the effective number of customers was reduced significantly. The authors furthermore used the concept of feasibility graph and concepts from computational geometry, such as Voronoi diagram, to develop an effective heuristic. But this article did not explicitly consider the stochastic nature of customer demand because the fixed routes were only designed to accommodate peak demand.

We also note that Campbell, Clarke, and Savelsbergh (2002) utilized a clustering idea for solving an Integer Programing model for their routing problem. Their main purpose for clustering customers was to reduce the number of routes by allowing customers to be on the same route only if they are in the same cluster. On any given day, the customers in the same cluster can still be assigned to different routes.

Compared to the standard VRP and SVRP research, little seems to have been published that explicitly measures driver learning. In this research, we explicitly consider the value of driver familiarity by introducing a learning curve model and develop concepts such as "cells", "core areas" and "flex-zones". Based on these models and concepts, a two-stage vehicle dispatching model is developed to balance the tradeoff of route optimality and driver familiarity with their service territory, hence dispatch consistency.

3. PROBLEM DESCRIPTION

Our research focuses on the transportation of freight within local regions. Routes have the characteristic that the driver returns to a home base at the end of each work period. The local portion of parcel systems and express mail systems (e.g. UPS and FedEx) are good examples. Typically, a local distribution network consists of multiple depots or centers. Each depot is associated with a geographic area as well as a fleet of vehicles, which is responsible for delivering packages destined to customers in this service area. The customers are spatially distributed and each day both the locations of customers that require service and their demand are stochastic.

In this paper, we consider a single depot problem (defined in Section 4.1), where the center management creates both strategic (period) routes and operational (daily) routes in separate steps, with the goal of minimizing total routing cost over multiple days, accounting for the number of drivers utilized, total distance traveled and total time to serve stops. This goal is achieved by constructing routes that are compact, efficient and flexible, while minimizing day to day variations in driver territories, and obeying constraints on customer demands and the maximum daily working duration for each driver. This problem will be challenging for the following reasons:

1) The time to serve a set of stops varies from driver to driver, and depends on how often, and when, each driver previously visited the stops.

2) Customer locations and demand are random, necessitating route adaptation each day.

3) Dispatch efficiency is judged both by the total cost of providing service and the quality of service offered to customers, as reflected in driver familiarity.

Our goals in the research are both to identify service territories and route "architectures" (i.e. network topologies) that perform well in practice, and to identify operational policies for dispatching vehicles that result in near optimal routes in practice. For simplicity, we make the following assumptions in this research:

1) The number of customers requesting delivery and their locations are revealed before vehicles are routed within each execution of our operational (daily routing) model, but are not known with certainty in our strategic (period routing) model since the execution of this model is performed before the demand is known.

2) Time windows are not explicitly considered.

3) Route planning is limited to deliveries only. We assume pickups are scheduled separately.

4) The vehicle capacity is not a constraining resource. That is, there is sufficient space in the vehicle for all packages on each route. However, each route is constrained by the length of the driver's work-shift.

5) Vehicles begin and end each day at a single depot.

3.1 Routing Concepts

Our routing methodology relies on several new concepts for characterizing vehicle routes, which we call "cells", "core areas" and "flex-zones" (see Figure 3.1.).

1) "Cell" is defined as the minimum unit of a service area whose whole workload is assigned to a single driver. Cells can be defined by grouping customers according to postal codes. The demand of each cell is defined as the summation of demands of all customers in the cell. Routing on the cell level has many advantages. First, by routing customers by cells, the problem size is reduced dramatically. This is of great importance in large-scale vehicle dispatching problems. Secondly, driver learning is more effectively modeled on the cell level than on the single stop level, because it is easier for a driver to become familiar with streets in a neighborhood than with individual customer stops. Throughout the paper, we use the term "stop" to refer to a single customer location and the term "cell" as a group of stops.

2) "Core area" is a group of cells that are served by the same driver every day, ensuring that a portion of each driver's route is stable from day to day. The core area is constructed within our strategic planning model, as will be explained in Section 4. In Figure 3.1, the darkest cell in each core area represents a possible seed point for the core area. The number of core areas corresponds to the minimum number of drivers used on any given day.

3) "Flex-zone" is a region around the depot that is deliberately excluded from core areas so that its stops can be reassigned daily. The flex-zone provides an efficient way to balance loads among routes each day, because many routes pass through it, providing many alternatives for assigning cells to routes within our

operational (daily) model. From Figure 3.1 it should be noted that cells in the flex-zone are excluded from core areas prior to application of our strategic model. After execution of the strategic model, additional cells, falling outside of the flex-zone, will also be unassigned to core areas. Both types of unassigned cells are assigned to routes on a daily basis within our operational model. When demand is low, unassigned cells are served by the minimum set of routes, represented by the set of core areas. When demand is high, additional routes are inserted to serve unassigned cells.

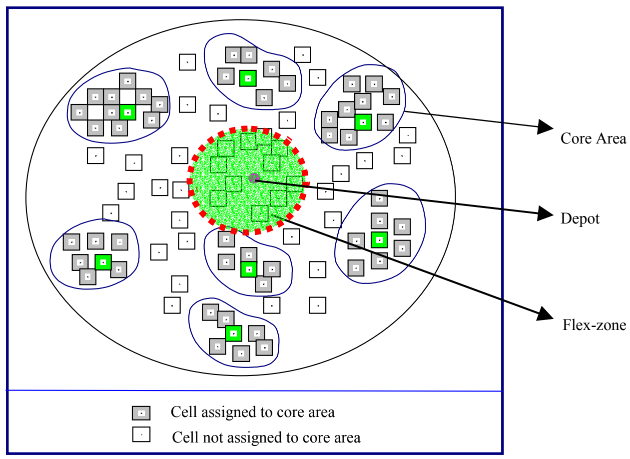

Figure 3.1. Illustration of Cell, Core Area and Flex-zone

3.2 Driver Learning Model

Our routing method explicitly models driver familiarity with learning curves (Hancock and Bayha, 1992). We utilized learning curves in which a driver's performance (in terms of time spent to finish the workload in a cell) is a function of the number of times he has visited the cell. A cell is used as the basic unit because driver learning is related to the driver's ability to navigate through neighborhoods. We also introduce a forgetting curve to model decreasing driver performance when a driver does not visit a cell for one or more days, which we call an interruption. For this purpose, we utilized the Variable Regression to Invariant Forgetting (VRIF) model by Elmaghraby (1990), which is based on the work of Wright (1936). The VRIF model incorporates both learning and forgetting curves mathematically and assumes that there is a unique forgetting function that intercepts the axis representing the time to produce a unit (later called $\hat{T}_1$).

As illustrated on Figure 3.2(a), as the numbers of visits to a cell by the same driver increases, the average time spent to serve each stop in this cell approaches the lower limit T_∞. A learning limit is reasonable because driver performance is limited by physical constraints on speed. The mathematical form for the driver learning curve is as follows: $T_j = \max\{T_1 j^{-l}, T_\infty\}$, where T_1 represents the average time to serve each stop (driving time plus time at stop) on the first visit to a cell and T_j represents the average time to serve each stop on the *j*th visit to a cell. *l* is the learning slope, a constant for any given situation. The learning limit will be reached when a driver is completely familiar with the cell.

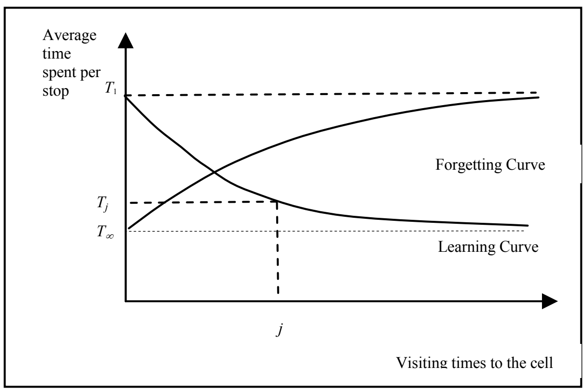

Figure 3.2(a) . Learning Curve with Forgetting

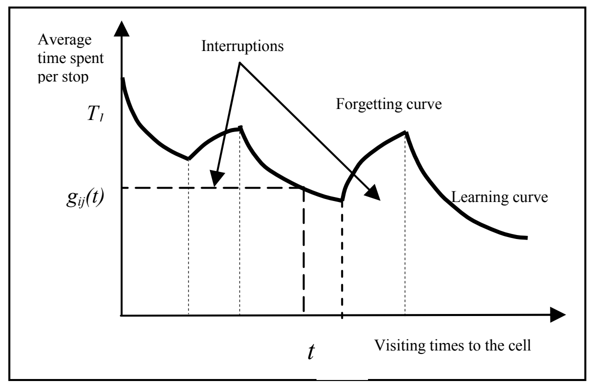

Figure 3.2(b). Dynamic Learning Function

The shape of the forgetting curve has similar form as the learning curve: $\hat{T}_x = \min\left\{\hat{T}_1 x^f, \hat{T}_\infty\right\}$, where *x* is the number of days that have elapsed since the last visit to the cell. We assume $\hat{T}_1 = T_\infty$ and $\hat{T}_\infty = T_1$ meaning that the starting point for the forgetting curve is the same as the learning limit. We also assume learning slope *l* and forgetting slope *f* are the same, so we have $0 < l = f < 1$. By tracking driver visits to cells, we can build a dynamic learning function with forgetting, $g_{ij}(t)$, representing the time for driver *i* to serve a stop in cell *j* on day *t*, as illustrated in Figure 3.2(b), which can be derived from $g_{ij}(t\text{-}1)$ in the following recursion:

$$g_{ij}(t \mid t-1) = \begin{cases} \max\{T_1 \cdot ((\frac{g_{ij}(t-1)}{T_1})^{-\frac{1}{f}} + 1)^{-f}, T_\infty\}, \text{if driver } i \text{ visits cell } j \text{ on day } t \\ \min\{T_\infty \cdot ((\frac{g_{ij}(t-1)}{T_\infty})^{\frac{1}{f}} + 1)^{f}, T_1\}, \text{ if driver } i \text{ interupts visiting cell } j \text{ on day } t \end{cases}$$

In this equation, $(\frac{g_{ij}(t-1)}{T_1})^{-\frac{1}{f}}$ and $(\frac{g_{ij}(t-1)}{T_\infty})^{\frac{1}{f}}$ represent the equivalent number of visits whose associated performance level equals to $g_{ij}(t\text{-}1)$ on the learning curve and forgetting curve, respectively. The total time for driver i to serve cell j on day t is then the product of the stop service time, $g_{ij}(t)$, and the number of stops that need to be served in cell j on day t. The dynamic learning function is important in our operational model, in which we assign cells to drivers on a daily basis, because different drivers assigned to the same cell on the same day would have different service times, due to variation in familiarity with the service territory.

Within our strategic model, we will also need to know the expected service time given that a driver visits the cell with a certain probability or frequency. If the probability that driver i visits cell j on any given day is p, then his performance level will increase according to $g_{ij}(t)$ with probability p each day, while his performance level will decrease according to $g_{ij}(t)$ with probability 1-p. Assume that driver performance level is x on day t-1. Then the expected performance level on day t is the following function of p:

$$\overline{g}_{ij}(t,p) = E[g_{ij}(t \mid g_{ij}(t-1) = x)] = p \cdot \max\{T_0 \cdot ((\frac{x}{T_0})^{-\frac{1}{f}} + 1)^{-f}, T_\infty\} + (1-p) \cdot \min\{T_\infty \cdot ((\frac{x}{T_\infty})^{\frac{1}{f}} + 1)^{f}, T_0\}$$

We proved in Zhong (2001) that $\overline{g}_{ij}(t,p)$ converges under independence, e.g.: $\lim_{t\to\infty} \overline{g}_{ij}(t,p) = \tilde{g}_{ij}(p)$. In simulation experiments we found that the function converges to its limit rather quickly, usually within 30 or 40 steps. We then take this limit as the expected driver performance level given a visiting frequency p of driver i to cell j in our strategic planning stage.

3.3 Two-stage Vehicle Dispatching Model

Our approach is to first design a series of core areas in a strategic model that serve as "a priori" service territories and then minimize total daily cost by utilizing a cell routing procedure in our operational model. By designing a priori service territories, drivers can become familiar with a large portion of their customers. By implementing a daily assignment of unassigned cells that are not part of a core area, dispatching is flexible, hence increasing efficiency and driver utilization.

1) Strategic Core Area Design (SCAD) stage

In the Strategic Core Area Design (SCAD) stage, a set of core areas is identified by solving a non-linear generalized assignment program using a tabu search heuristic with the objective of minimizing the cost associated with assigning a cell to a core area, which will be defined later. We use tabu search because the non-linear generalized assignment problem is a NP-hard problem thus finding optimal solutions for this problem is computational prohibitive for reasonably sized problems, and tabu search has been shown to be an effective method for solving this type of problem (Laguna and *al*, 1995). The assignment is constrained by a threshold that is the probability that the total workload (in terms of total service and travel time) in the core area exceeds the working duration of a driver.

2) Operational Cell Routing stage

This stage is where the core area design solution is implemented on a daily basis by building cell tours. First a partial cell tour is built among the cells in each core area. Then the cells that are not pre-assigned to any core area, including the cells in the flex-zone, are added to these partial cell routes at the lowest cost and new cell tours are added if necessary. In this stage we implement a cell routing-with-learning schema, which introduces a learning curve model as well as a continuous spatial model for estimating the workload within a cell and the cell-to-cell travel distance. The solution method for this stage is based on a modification of existing Vehicle Routing algorithms.

4. STRATEGIC CORE AREA DESIGN MODEL

In the strategic model, each cell is either assigned to a core area or left unassigned. However, before core area construction, a certain percentage of the cells are assigned to a flex-zone. This percentage is an input to the model and the cells closest to the depot are selected to be in the flex-zone. Then, the remaining cells are considered for assignment to a core area. For formulation convenience, all cells considered during the core area construction phase that do not become part of a core area are assigned to dummy core area *0*. In the strategic planning stage, we do not construct cell tours because actual cell tours will vary from day to day, as determined by the operational model.

The objective of the strategic problem is to minimize the expected total cost (computed as the expected total time to satisfy the demand in terms of vehicle travel and service time) associated with the assignment of cells to core areas, under the constraint that limits the probability that the sum of the workload (measured as the total travel and service time) within each core area exceeds the maximum work duration of drivers. In this paper, we used a fixed number of core areas, which would be the minimum number of drivers used from historical data over a certain period.

4.1. General Formulation

Let X represent the set that contains the n cells that need to be served and X_k represents the set of cells that are assigned to core area $k\ (k = 0,1,..., m)$. Here, when $k \neq 0$ X_k represents all the cells that are assigned to core area k, while X_0 represents all the cells that are left unassigned (i.e. they are assigned to dummy core area *0*).

Define $W(X_k)$ as the total workload for core area k. Then our problem can be formulated as:

$$Min \quad E\left[\sum_{k=0}^{m} W(X_k)\right] \tag{1'}$$

$$S.t. \quad P\left(W(X_k) \leq Q_k\right) \geq 1-\alpha, \quad (k=1,..., m) \tag{2'}$$

$$\bigcup_{k=0}^{m} X_k = X, \tag{3'}$$

$$X_{k_1} \cap X_{k_2} = \phi \qquad (k_1, k_2 = 0,1,..., m \text{ and } k_1 \neq k_2) \tag{4'}$$

where the parameters of the model are as follows:

Q_k = Maximum working duration for driver k

α = Threshold probability that the total workload in each core area can exceed the maximum working duration.

However, since we can not construct an exact cell sequence in this strategic level, it is hard to create an exact formula for $W(X_k)$. Our method is to use an approximation based on cell to core area assignment without actually constructing cell tours. Let I_{ik} equal to one when cell i is assigned to core area k. Then the total time associated with assignment I_{ik} is: $f_{ik} = e_{ik} + h_{ik}$, where e_{ik} is the total time for the driver serving core area k to finish the workload within cell i. h_{ik} is the total time contribution of all cell-to-cell travel of assignment I_{ik} in the cell tour that covers core area k. Now, we have $W(X_k) = \sum_{i=1}^{n}(e_{ik} + h_{ik})I_{ik}$. And our model becomes:

$$Min \qquad E\left[\sum_{i,k}(e_{ik} + h_{ik})I_{ik}\right] \qquad (1'')$$

$$S.t. \qquad \sum_{k=0}^{m} I_{ik} = 1, \qquad i = 1,2,\ldots, n \qquad (2'')$$

$$P\left(\sum_{i}(e_{ik} + h_{ik})I_{ik} \le Q_k\right) \ge 1-\alpha, \qquad (k=1,\ldots,m) \qquad (3'')$$

$$I_{ik} = 0 \; or \; 1, \qquad \begin{matrix} i = 1,2,\ldots, n \\ k = 0,1,\ldots m \end{matrix} \qquad (4'')$$

Again, because we can not construct the exact cell sequence in this stage, e_{ik} and h_{ik} become extremely complicated functions and cannot be expressed for nontrivial problems. Our heuristic method is based on constructing linear approximations: $e_{ik} = \rho_{ik} T_i \xi_i$ and $h_{ik} = \rho_{ik} C_{ik}$, where ρ_{ik} is the learning factor of driver k in cell i, T_i is the average time needed to serve a single stop in cell i (includes the individual stop-to-stop travel time and stop service time but does not include cell-to-cell travel time), ξ_i is the number of customer stops in cell i (random variable) and C_{ik} is the cost of assigning cell i to core area k (in terms of cell-to-cell travel time in a cell tour). Note that the cell-to-cell travel time is based on the seed point of core area k rather than the cell sequence because an exact sequence is not constructed during the core area design level in order for the model to be computationally tractable.

As we can see, the value of consistency or familiarity is reflected in the parameter ρ_{ik}. The basic assumption is that as the driver visits a cell more often the driver will be more familiar with its location. If the driver is more familiar with a cell, he or she will need less time to finish the workload within this cell as well as less time to travel from cell-to-cell in the cell tour.

For the assignment cost C_{ik}, we have two scenarios to consider: $k \ne 0$ and $k = 0$.

1. In the case of $k \ne 0$, the learning factor ρ_{ik} is a pre-defined value. It is the learning limit of driver k in cell i. We assume that this value is identical for all the cells if they are assigned to core areas.

The standard cell-to-cell travel time contribution of assignment I_{ik} in a cell tour is estimated as

$$C_{ik} = d_{ik} + d_{i0} - d_{ok}$$

where d_{ik} = Expected travel time from cell i to the seed point of core area k.

d_{i0} = Expected travel time from cell i to the dummy core area 0 (depot used as the seed point).

d_{0k} = Expected travel time from depot to seed point of core area k.

In our experiments, the seed points of core areas were obtained by solving a m-median problem for all centroids of cells (in real applications, seed points may be based on service factors or observed clusters of customers). Fisher and Jaikumar (1981) used a similar method to approximate the contribution of assigning a single stop to a certain route. Their heuristic was based on the idea of assigning customers to routes first and sequencing customers later. They then used a similar estimation to determine the extra distance traveled if a stop is assigned to an existing route. The authors reported good approximation results and we will see in a later section that this method is also suitable for our problem. We found that C_{ik} is a rather large overestimation of what it should really be. Our method is to adjust the value by a multiplier which was determined through experimentation.

2. In the case of $k = 0$ -- that is, if the cell is left unassigned -- flexibility is measured from visiting frequency. If a cell is left outside of all the core areas, it will be assigned to different drivers on a daily basis. Assume that the frequency that cell i is assigned to core area k ($k \neq 0$) is F_{ik} . We then know from section 3.2 that the expected learning performance level of driver k in cell i is $\tilde{g}_{ki}(F_{ik})$. Then the expected learning curve factor in cell i is $\rho_{i0} = \sum_{1}^{m} F_{ik}\tilde{g}_{ki}(F_{ik})$.

From this equation, we know that if a cell is split among several drivers in the daily dispatching stage, the expected learning curve factor will be greater than that of a cell that is assigned to a core area. This means that there will be less of a learning benefit if a cell is left unassigned in the core area building process. However, unassigned cells in the daily dispatching stage provide more freedom for the dispatch system to balance the workload and efficiently construct daily routes. The benefits resulting from this flexibility can not be explicitly expressed as a direct functional relationship. We indirectly measure the value of this flexibility by calculating the expected cell-to-cell travel time contribution C_{i0} of unassigned cell *i*:

$$C_{i0} = (\frac{d_{ib_2} - d_{ib_1}}{d_{ib_2}}) \cdot (F_{ib_1} C_{ib_1} + F_{ib_2} C_{ib_2}) + \frac{d_{i0}}{\bar{d}_{b0}} F_{i0} \cdot d_{i0},$$

where the core areas are ranked by the increasing distance from cell *i* and b_1, b_2 are the indices of the core areas with the best and second best rank. We now define the variable F_{i0} as the frequency that cell *i* is not assigned to either b_2 or b_2. Here, $\bar{d}_{b0} = \frac{d_{b_1 0} + d_{b_2 0}}{2}$. Note that d_{ib_1} is the expected travel time of cell *i* to core area b_1 and similarly we use the depot as the seed for the computation of d_{i0}.

We have considered the flexibility in two dimensions: one dimension is towards the depot, and the other one is towards neighboring driver territory. As we earlier mentioned, a cell that is not assigned to a core area provides the dispatcher the flexibility in efficiently assigning that cell to a driver. This flexibility is greater when the cell is closer to the depot and more equally positioned between two neighboring drivers. In the definition of C_{i0}, $(\frac{d_{ib_2} - d_{ib_1}}{d_{ib_2}})$ and $\frac{d_{i0}}{\bar{d}_{b0}}$ serve as the *flexibility coefficients* in these two dimensions respectively. We can see that those cells that have nearly equal distance as the two closest core areas and those that are close to the depot have smaller values of C_{i0}, which shows that those cells have more flexibility. This method was used to indirectly approximate flexibility.

For the sake of consistency, we modify the definition of ρ_{i0} as follows:

$$\rho_{i0} = F_{ib_1} \tilde{g}_{b_1 i}(F_{ib_1}) + F_{ib_2} \tilde{g}_{b_2 i}(F_{ib_2}) + F_{i0} \tilde{g}_{0i}(F_{i0})$$

We now can clearly see the tradeoff between familiarity and flexibility. When a cell is assigned to one core area, it will benefit from maximum learning, which is reflected in the learning curve factor ρ_{ik}. When it is left unassigned, it benefits from the flexibility of being assigned to different drivers each day which is reflected in C_{i0}.

4.2. Stochastic Program Formulation

This section completes the primal stochastic program formulation for our Strategic Core Area Design model, which is formulated as follows:

$$Min \quad E\left[\sum_{i,k} \rho_{ik}\, T_i\, \xi_i I_{ik} + \sum_{i,k} \rho_{ik} C_{ik} I_{ik}\right] \tag{1}$$

$$S.t. \quad \sum_{k=0}^{m} I_{ik} = 1, \qquad i = 1,2,\ldots, n \tag{2}$$

$$P\left(\sum_i (\rho_{ik} T_i \xi_i + \rho_{ik} C_{ik}) I_{ik} \le Q_k\right) \ge 1-\alpha, (k=1,\ldots,m) \tag{3}$$

$$I_{ik} = 0 \; or \; 1, \qquad \begin{array}{l} i = 1,2,\ldots, n \\ k = 0,1,\ldots m \end{array} \tag{4}$$

The objective function (Eq. 1) is the sum of the costs for assigning cells to the core areas (including dummy core area *0*). $\rho_{ik}\, T_i\, \xi_i$ in the objective function represents the learning-adjusted workload within the cell (including stop-to-stop travel time and service time at each stop). $\rho_{ik} C_{ik}$ represents the contribution of learning-adjusted cell-to-cell travel time in the cell tour based on the cell assignment. Eqs. 2 ensure that each cell is assigned to only one core area. Eqs. 3 are the probability constraints for the working duration of each core area. It states that the probability that the total workload in each core area exceeds the maximum working duration of the driver cannot be higher than α.

4.3. SCAD Solution Method

We assume the number of customer deliveries in each cell *i*, ξ_i, are independent normally distributed random variables with means μ_i and standard deviations σ_i. We denote M_k and S_k as the mean and standard deviation of the workload in core area k, respectively, and $M_k = \sum_i \rho_{ik} T_i \mu_i I_{ik}$, $S_k = \sqrt{\sum_i (\rho_{ik} T_i \sigma_i I_{ik})^2}$. Stewart and Golden (1983) proved that if ξ_i is normally distributed, there exists a constant τ such that, $\Pr\left[\left(\sum_i \rho_{ik} T_i \xi_i I_{ik} - M_k\right) \Big/ S_k \le \tau\right] = 1-\alpha.$

Therefore, the chance constraint (3) becomes: $\sum_i \rho_{ik} T_i \mu_i I_{ik} + \tau \sqrt{\sum_i (\rho_{ik} T_i \sigma_i I_{ik})^2} \leq Q_k - \sum_i \rho_{ik} C_{ik} I_{ik}$.

And the chance-constrained program becomes a non-linear integer program:

$$Min \quad E\left[\sum_{i,k} \rho_{ik}\, T_i\ \xi_i I_{ik} + \sum_{i,k} \rho_{ik} C_{ik} I_{ik}\right] \tag{5}$$

$$S.t. \quad \sum_{k=0}^{m} I_{ik} = 1, \qquad i = 1,2,....,n \tag{6}$$

$$\sum_i \rho_{ik} T_i \mu_i I_{ik} + \tau \sqrt{\sum_i (\rho_{ik} T_i \sigma_i I_{ik})^2} + \sum_i \rho_{ik} C_i I_{ik} \leq Q_k, \ \ (k=1,...,m) \tag{7}$$

$$I_{ik} = 0\, or\, 1, \qquad \begin{array}{l} i = 1,2,....,n \\ k = 0,1,....m \end{array} \tag{8}$$

Notice that: $E\left[\sum_{i,k} \rho_{ik}\, T_i\ \xi_i I_{ik} + \sum_{i,k} \rho_{ik} C_{ik} I_{ik}\right] = \sum_{i,k} E[\rho_{ik}\, T_i\ \xi_i I_{ik}] + \sum_{i,k} \rho_{ik} C_{ik} I_{ik} = \sum_{i,k} \rho_{ik}\, T_i\ \mu_i I_{ik} + \sum_{i,k} \rho_{ik} C_{ik} I_{ik}$

So, the above model becomes:

$$Min \quad \sum_{i,k} \rho_{ik}\, T_i\ \mu_i I_{ik} + \sum_{i,k} \rho_{ik} C_{ik} I_{ik} \tag{9}$$

$$S.t. \quad \sum_{k=0}^{m} I_{ik} = 1, \qquad i = 1,2,....,n \tag{10}$$

$$\sum_i \rho_{ik} T_i \mu_i I_{ik} + \tau \sqrt{\sum_i (\rho_{ik} T_i \sigma_i I_{ik})^2} + \sum_i \rho_{ik} C_{ik} I_{ik} \leq Q_k, \ \ (k=1,...,m) \tag{11}$$

$$I_{ik} = 0\, or\, 1, \qquad \begin{array}{l} i = 1,2,....,n \\ k = 0,1,....m \end{array} \tag{12}$$

Our approach to solve this Non-linear Generalized Assignment Problem (NGAP) is to develop a tabu search heuristic. The tabu search method was based on the work of Laguna and *al* (1995). They developed a tabu search heuristic to solve the Multilevel Generalized Assignment Problem (MGAP), which differs from GAP in that agents can perform tasks at more than one efficiency level. Our tabu search implementation extends their approach by incorporating the non-linear constraints and modifying the method for constructing ejection chains and dynamic tabu tenures. We briefly outline our tabu search method next.

In tabu search, a *move* is defined as a modification to a current solution according to some predefined procedures so as to form another new solution. The collection of solutions that result from all possible moves around the current solution is called a neighborhood of the current solution. Based on a current solution of NGAP, a basic move is to simply change the assignment of cell i from core area j to core area k. Notice that after every move, we always satisfy the assignment constraints (10), but the capacity constraints (11) can not be guaranteed to hold. In another words, infeasibility may occur due to the violation of the capacity constraints. This special feature of the move enables the tabu search to cross the feasible region and lead to possible global optimal solution.

Contrasted to the simple move above, a *compound move* refers to a combination of a series of simple moves, whose number of components is decided by a certain compound move construction termination procedure. Compound moves do not concern isolated choices of moves with particular features, but rather involve coordinated choices of moves with interlinking properties. Procedures that incorporate compound moves are often called *variable depth methods*, based on the fact that the number of components of a compound move generally varies from step to step. A neighborhood constructed by a compound move is more likely to produce effective and efficient moves.

An *Ejection chain* is a special subclass of the class of variable depth methods. According to Glover and Laguna (1997), an ejection chain is started by selecting a set of elements, whose states will undergo some changes (e.g. to new values or new positions). Consequences of this change will lead to at least one element in the set to be "ejected" from their current state. Figure 4.1 shows an example ejection chain for our NGAP problem. From the illustration, dashed lines represent the original assignment: cell i_1 to core area j_1, i_2 to j_2 and so on. A simple move here is to reassign i_1 to core area j_2. An ejection chain results by allowing this move to eject an element - arc (i_2, j_2) at node j_2, whose node i_2 must be reassigned to a new core area, say j_3. This process may continue through additional nodes until a suitable termination criterion is met. The simple termination criterion for our problem could be continuing the chain until no ejection occurs. This is possible because some core area down the chain might have enough capacity to serve the last ejected node (at least dummy core area *0* has sufficient since it is not capacity constrained).

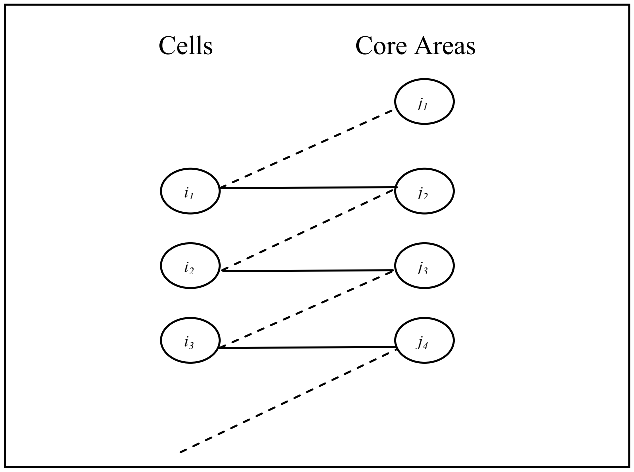

Figure 4.1. Ejection Chain Illustration in NGAP

Before giving the method used in our implementation to construct an ejection chain for our NGAP problem, let's first define the capacity slack of core area *k,* $s_k = Q_k - \sum_i \rho_{ik} T_i \mu_i I_{ik} - \tau \sqrt{\sum_i (\rho_{ik} T_i \delta_i I_{ik})^2} - \sum_i \rho_{ik} C_{ik}$.

A solution is feasible only if all $s_k \geq 0$. We use the notation in Laguna et al. (1995) to define a measure of infeasibility, v, of an assignment-feasible solution *x* as the absolute value of the sum of all the negative capacity slacks, i.e., $v(x) = abs\ (\sum_{k=1}^{m} \min(s_k, 0))$. Therefore, solution *x* is feasible only if $v(x) = 0$. Figure 4.2 shows the graphical representation of the ejection chains we used. Table 4.1 provides a summary of the three types of ejection chains for our NGAP problem.

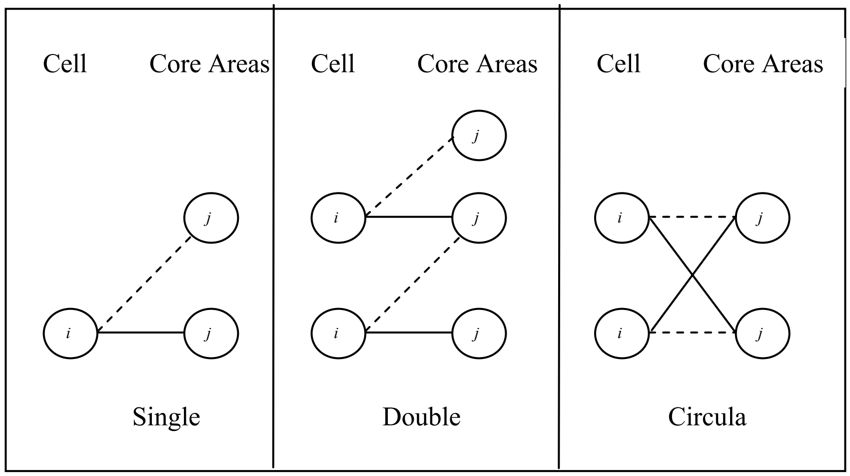

Figure 4.2. Ejection Applied in NGAP

Type	Leaving arcs	Entering arcs	Feasibility condition
Single	(i, j_1)	(i, j_2)	$s_{j_2} - (\rho_{ij_2} T_i \mu_i + \rho_{ij_2} C_{ij_2} + \phi_{ij_2}) \geq 0$
Double	(i_1, j_1) (i_2, j_2)	(i_1, j_2) (i_2, j_3)	$s_{j_2} + (\rho_{i_2 j_2} T_{i_2} \mu_{i_2} + \rho_{i_2 j_2} C_{i_2 j_2} + \sigma_{i_2 j_2}) - (\rho_{i_1 j_2} T_{i_1} \mu_{i_1} + \rho_{i_1 j_2} C_{i_1 j_2} + \phi_{i_1 j_2}) \geq 0$ $s_{j_3} - (\rho_{i_2 j_3} T_{i_2} \mu_{i_2} + \rho_{i_2 j_3} C_{i_2 j_3} + \phi_{i_2 j_3}) \geq 0$
Circular	(i_1, j_1) (i_2, j_2)	(i_1, j_2) (i_2, j_1)	$s_{j_1} + (\rho_{i_1 j_1} T_{i_1} \mu_{i_1} + \rho_{i_1 j_1} C_{i_1 j_2} + \sigma_{i_1 j_1}) - (\rho_{i_2 j_1} T_{i_2} \mu_{i_2} + \rho_{i_2 j_1} C_{i_2 j_1} + \phi_{i_2 j_1}) \geq 0$ $s_{j_2} + (\rho_{i_2 j_2} T_{i_2} \mu_{i_2} + \rho_{i_2 j_2} C_{i_2 j_2} + \sigma_{i_2 j_2}) - (\rho_{i_1 j_2} T_{i_1} \mu_{i_1} + \rho_{i_1 j_2} C_{i_1 j_2} + \phi_{i_1 j_2}) \geq 0$

Table 4.1. Ejection Chains and Their Feasibility Conditions

We next describe a method to develop a lower bound for the NGAP in order to assess the quality of the solution given by our tabu search approach. The nonlinear generalized assignment problem, especially with nonlinear constraints, is computationally prohibitive to optimally solve. Our lower bound approach is to linearize the constraint (11). We take the following equations as the linearized counterparts of the nonlinear constraints (11):

$$\sum_i \rho_{ik} T_i \mu_i I_{ik} + \frac{\tau \sum_i (\rho_{ik} T_i \sigma_i I_{ik})}{\sqrt{n}} + \sum_i \rho_{ik} C_{ik} I_{ik} \leq Q_k, \quad (k = 1, \ldots, m) \qquad (13)$$

Notice that if equations (11) are met, then equations (13) are also met. This means that the feasible region restricted by constraints (13) contains the feasible region restricted by constraints (11). Therefore, the linear GAP problem formed by (9), (10), (13), and (12) must contain the optimal solution of our NGAP problem formed by

(9), (10), (11), and (12). Hence, the optimal solution of the approximation GAP problem can serve as the lower bound of our NGAP problem.

Note that even the linear approximation model of the NGAP is still a mixed integer linear program. However, it is easier to computationally solve using a commercial software package than the NGAP. We used the CPLEX MIP Solver to solve the linear approximation model in order to obtain a lower bound. We then benchmark the tabu search solution for the NGAP against the best lower bound found by CPLEX on the linearized model. In section 6.2, we show the effectiveness of our proposed tabu search solution by showing that the gap between its solution and the lower bound is small on a number of experiments.

4. OPERATIONAL CELL ROUTING MODEL

After the Strategic Core Area Design stage, we have identified a set of core areas that will serve as drivers' core service territories. At the operational level, the cells that are not assigned to any core area will be assigned to one of these drivers or extra drivers are added if needed. The dispatch system also needs to determine the cell sequence, which drivers can follow to make the deliveries.

The problem in this stage (Figure 5.1.) is to design a set of least-time duration vehicle routes such that:

1) Each cell excluding the depot is visited exactly once.
2) All vehicle routes start and end at the depot.
3) Total on-road time duration restrictions are satisfied on all vehicle routes.
4) All cells that are pre-assigned to a vehicle/driver are served by the same vehicle/driver.

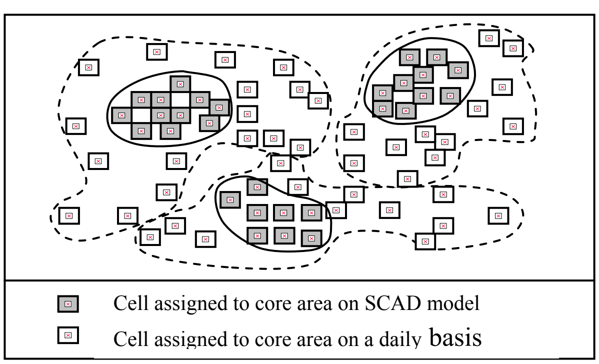

Figure 5.1. Operational Cell Routing Model

The operational cell routing model is very similar to the classical VRP with maximum driver work duration constraints (Fisher and Jaikumar, 1981). The major difference is the pre-assignment constraint. Based on this fact, we developed a modified VRP algorithm. However, our problem is more challenging because:

1) We are routing cells instead of individual stops. Routing based on the cell level is more complicated, since the time required to serve a cell depends on the preceding and following cells (as these affect the first and last stop visited in the cell)

2) The driver learning effect needs to be incorporated in the route construction process, which tends to assign cells to more familiar drivers.

Our cell routing method uses an algorithm developed at UPS (Zaret, 1999) which is based on a parallel insertion heuristic (see e.g. Potvin and Rousseau, 1993). It was modified to account for the two above challenges. The core areas given by the SCAD model provide the number of drivers, *k*, and the pre-assignment of some cells to a core area. Therefore, as a starting point, we already have *k* partial cell tours.

First, let's see how to route cells instead of single stop. On the operational level, each day the driver needs to start out from the depot and drive to the first cell, finish all the deliveries within this cell and go to another cell, and so on. Thus, the whole cell could be viewed as a "super stop", whose longitude and latitude could be the centroid of all the stops within this cell and service time the total time needed to serve all the stops in the cell, including the travel time among the stops and the service time at each stop. This "super stop" service time can be approximated using the method described in Section 4. Now we have a location and service time for the "super stop". We then need a method to define the insertion cost, which inserts one cell into the cell tour.

Assume the current cell tour consists of cells $(1,\ldots, i, j,\ldots, n)$ in that order. u is an unrouted cell, then the cost of inserting u between cell i and j is:

$$C(i,u,j) = d_{iu} + d_{uj} - d_{ij}$$

Second, driver learning is incorporated in the route construction procedure as follows. Our method is to build a dynamic driver performance matrix *Pmat*, which stores all drivers' performance level in all cells, in terms of the percentage of time it needs to serve the cells or traveling from cell-to-cell. Then *Pmat (i, k)* is the current performance level for driver k in cell i. After the dispatching plan is made at the beginning of each day, this matrix *Pmat* needs to be updated according to the cell assignment.

If cell i is assigned to driver k again, then according to learning curve, *Pmat (i, k)* should be updated to:

$$Pmat(i,k) = \max\{T_0 \cdot ((\frac{Pmat(i,k)}{T_0})^{-\frac{1}{f}} + 1)^{-f}, T_\infty\}$$

where T_0 is the starting performance level, f is the learning and forgetting rate, and T_∞ is the learning limit. When we start from $T_0 = 100\%$, then we have:

$$Pmat(i,k) = \max\{((Pmat(i,k))^{-\frac{1}{f}} + 1)^{-f}, T_\infty\}$$

If cell i is assigned to a driver other than k, then according to forgetting curve, *Pmat (i, k)* should be updated to:

$$Pmat(i,k) = \min\{T_\infty \cdot ((\frac{x}{T_\infty})^{\frac{1}{f}} + 1)^{f}, T_0\}$$

This dynamic driver performance matrix *Pmat* serves as a multiplier for the total time needed to serve all stops in cell i by driver k, and the travel time from cell i to cell j.

The solution of the Operational Cell Routing Stage gives us a set of cell routes that will guide the drivers in serving the customers from cell-to-cell. In practice, the dispatching process can stop at the cell level and let the driver organize the stop sequence within the cells. However, in order to test our cell model against the traditional approach of no core areas, we need to determine the optimal stop sequence from a given cell sequence. This problem is referred to as the Stop Sequence Problem (SSP).

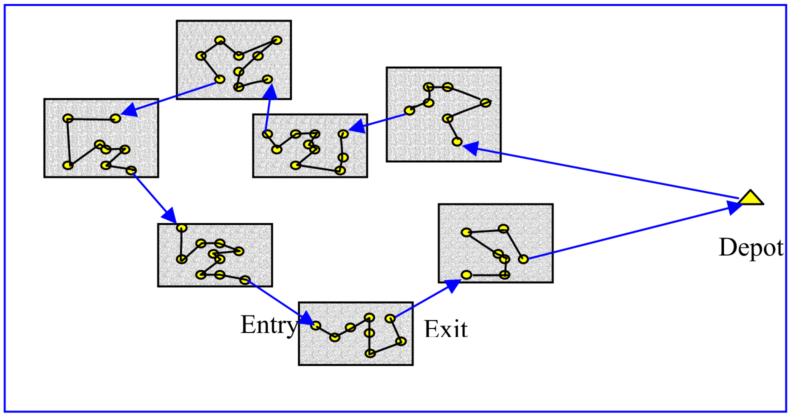

Figure 5.2. Demonstration of Stop Sequence Problem

The SSP assumes that nodes of a given graph *G* have been grouped into *m* mutually exclusive and exhaustive node subsets and all nodes need to be visited subset by subset in the given subset sequence as shown in Figure 5.2. There are two parts to the SSP. The first part determines the entry and exit stops for each cell as shown in Figure 5.2. This problem can be formulated as a Multistage Graph Problem (MGP). We optimally solved the MGP as a dynamic programming model. With the entry and exit know, the stop sequence within each cell is formulated as an Open Traveling Salesman Problem (OTSP). We again use dynamic programming to optimally solve the OTSP.

5. EXPERIMENTAL RESULTS

This section describes the methods used to measure the overall performance of our two-stage vehicle dispatching system.

6.1. Test Data Generation

Our test was based on a 30-day planning horizon. We tested our integrated two-stage vehicle dispatch model on an idealized delivery data set. Suppose that the service region ***A*** is a square area in the plane and the depot is located in the center. Transportation distance between points is assumed to be approximated by the Euclidean distance function.

We generate benchmark data in a similar manner as described in Solomon (1987). The customer stops are generated from three basic types: random uniform distribution (R type), clustered (C type) and semi-clustered (RC type). Semi-clustered data contain a mix of uniformly generated random data and clusters. We focus here on the RC type data for testing. RC data are more realistic, as R type data can represent residential customers and C type can represent business customers (which tend to be more clustered). Residential customers normally are scattered around the service region while business customers usually form clusters of different sizes. Moreover, residential customers and business customers have different probabilities for requesting service. Normally a business customer is more likely to request service and needs a longer service time. We note that we also ran experiments on just R type data and results were similar to those of the RC type so in this paper we only report the results for the RC type experiments.

We varied the number of individual customer stops from 100, 250, 500, 1000, 1500 and 2000. We assume that on average 70% of customers are residential and 30% are business. Associated with each customer stop is the probability of requesting service in each day. We assume a business customer has 80% probability of requesting service while a residential customer has 20% probability of requesting service in each day. We generated 10 problem sets for each problem size.

Once the potential customers in service region ***A*** are identified, we grouped them into cells. Our approach was to solve a *P-median problem* for all the potential customers and group customers to their closest medians. All customers grouped into the same median form a cell (in actual applications, cells would likely be based on geographic or postal code boundaries).

In the strategic planning stage problem, we used the following preset determined system parameters: driver learning limit = 90%, learning and forgetting rate = 95%, capacity threshold $\tau = 2.0$, and flex-zone size = 10%. Recall, the flex-zone size is the percentage of cells that will not be considered for assignment during the core area construction phase in order to provide the dispatcher flexibility to add cells to the different drivers to balance the workload on any given day.

For the operational cell routing stage, we ran a 30-day planning period simulation. On each day, we simulate the customer demand according to the delivery request probability using a Monte Carlo method. We then run the cell routing algorithm, which builds the cell tours. We then simulate how the driver performs within the cells by optimally solving a Stop Sequence Problem, which finds the optimal stop sequence for a given cell tour. The measurement criteria for the experimental tests are:

1. Number of drivers utilized
2. Total route duration (and average workload overage as described below)
3. Total route distance
4. Percentage of customers in core areas
5. Average visiting frequency of the highest frequency drivers for all cells

The total duration is the learning-adjusted total route duration. The duration of the stop sequence may exceed the maximum work duration limit. The average workload overage is a proper measure of this error. The average workload overage is the percentage of time spent by drivers in excess of their work shift duration (a measure of overtime). There almost always exists estimation error in approximating the workload within a cell and cell-to-cell distance, and this error will possibly propagate to the total duration of the stop sequence built from the cell tours.

The visiting frequency is the percentage of days that a particular driver has visited the cell. The driver who has the highest visiting frequency is called the *highest frequency driver* for the cell. This criterion measures the consistency of the dispatching plans.

6.2. NGAP Algorithm Results

Recall that we formulate our stochastic core area design stage as a stochastic program and solve it as a Nonlinear Generalized Assignment Problem (NGAP). We proposed a tabu search heuristic for solving the NGAP and a linearized GAP for bounding it. To obtain the solution of the linearized GAP, we used a CPLEX MIP solver. This solution serves as a lower bound for our NGAP. Because our CPLEX server had a node limit of 50,000, when the node limit was reached, we used the LP solution of the GAP as a lower bound for the associated NGAP. This situation is marked with " * " in the following tables.

We tested our tabu search method on 10 problem sets, which were based on randomly generating 500 customers based on the distributions given in Section 6.1. We then formed 100 cells from these customers. Table 6.1 shows an example problem set with its related average and standard deviation of number of customers and average service time needed in each cell. Test results showed that our tabu search heuristic performed very well compared to the lower bound. On average, the tabu solution is just 3% above the lower bound, which is shown in Table 6.2.

Cell i	μ_i	δ_i	T_i
1	3.866	0.902	5.997
2	5.556	0.970	6.373
3	5.062	1.072	5.428
4	4.037	0.783	5.562
5	1.632	0.860	5.363
6	3.568	0.941	5.657
7	2.268	0.695	5.999
8	1.839	0.746	5.457
9	3.912	0.837	6.628
10	4.416	0.986	6.124
…	…	…	…
100	2.175	0.693	6.748

Table 6.1. Customer Demand Parameters in Each Cell

Problem Set	**NGAP_Tabu**	**GAP_LB**	**% above Lower Bound**
1	2426	2346*	3.41%
2	2461	2395	2.76%
3	2446	2363*	3.51%
4	2406	2334*	3.08%
5	2391	2322	2.97%
6	2423	2356*	2.84%
7	2372	2311	2.64%
8	2431	2366	2.75%
9	2433	2352	3.44%
10	2412	2349	2.68%
Average	**2420**	**2349**	**3.01%**

Table 6.2. NGAP Solution Quality

We obtained similar results when the solution was tested under varying values of the learning curve factor, learning limit T_∞ and capacity threshold.

6.3. Value of Core Areas

Our method is to measure the value of the core areas by comparing the performance of our core area method with a "no-core-area" method, which entails re-optimizing cell routes on a daily basis. The no-core-area method has the most flexibility in reconstructing dynamic cell routes each day according to the changing demand in each cell. However, because of this maximum flexibility, each cell has the potential to be serviced from a different driver each day. Hence, driver learning at this cell will not approach the maximum learning level, e.g. the learning limit. In fact, the learning level is adjusted dynamically by the visiting frequency we previously mentioned.

In order to have a fair comparison, we deploy a "route-first, assign driver-later" approach for the no-core-area method as follows. After creating the cell routes, drivers are assigned to routes with the objective of minimizing the total route duration. To do this, we track driver performance in each cell according to the dynamic learning function, so the time to complete a route depends on the driver assigned to the route. This is a standard assignment problem. We implemented the Hungarian method developed by Kuhn (1955) to solve this problem with the assignment cost equal to the time required by an individual driver to complete a route. Now that we have incorporated the driver learning in the no-core-area method, we can compare the performance of these two approaches. We tested the core-area method and the no-core-area method on various problem sizes with the parameters generated by the methods described in Section 6.1. Tables 6.3 and 6.4 show the simulation results. The results are the average of the 30 days of the simulation.

Problem Size	Number Drivers	Total Duration	Total Distance	Total Overage	Frequency of Highest Frequency Driver
100 stops	2.00	750	156	2.14%	69.2%
250 stops	3.00	1517	263	1.63%	78.8%
500 stops	5.03	2719	421	1.11%	85.2%
1000 stops	9.21	4898	644	0.50%	79.3%
1500 stops	13.03	7023	838	0.45%	78.7%
2000 stops	17.00	9141	984	0.62%	76.6%

Table 6.3. Performance of the Core Area Method on data of different size.

Problem Size	Number Drivers	Total Duration	Total Distance	Total Overage	Frequency of Highest Frequency Driver
100 stops	2.00	748	149	2.34%	64.20%
250 stops	3.13	1572	250	1.38%	54.20%
500 stops	5.56	2835	412	1.42%	61.30%
1000 stops	9.67	5126	633	0.20%	61.60%
1500 stops	13.36	7319	829	0.25%	59.50%
2000 stops	17.30	9505	958	0.54%	62.80%

Table 6.4. Performance of the No-Core-Area Method on data of different size.

As we can see from the above tables, on average the core area method uses 4% fewer drivers and incurs 4% less total duration, while maintaining a 78% visiting frequency for the highest frequency drivers, which is 28% higher than that of the no-core method -- hence more consistency is provided. Thus, beyond providing less expensive routes, the core area method provides more consistent (and likely better) service for customers.

Note that the driver learning level at each cell plays an important role in the above two methods. Varying driver learning levels leads to varying workload in each cell and cell-to-cell travel time, hence varying total route durations. To create a lower bound for comparison, we now assume that all drivers are totally familiar with all cells in the no-core method, i.e., the learning levels of all drivers in all cells have reached the learning limit, at the beginning of our 30-day simulation. Then the solutions of this revised no-core-area method provides a lower bound on route performance (because it reoptimizes under the most favorable conditions each day). This is because the revised method has both the maximum flexibility and the maximum learning benefits. Based on this idea, we re-ran the revised no-core-area method on the same 10 problem sets and obtained the lower bound as illustrated in Table 6.5 below.

Problem Size	Number Drivers	Total Duration	Total Distance
100 stops	1.70	664	145
250 stops	2.93	1459	243
500 stops	4.81	2473	395
1000 stops	8.42	4519	586
1500 stops	12.03	6875	789
2000 stops	16.20	8199	962

Table 6.5. Lower Bound of the Core Area Method on data of different size.

Compared to this table, our core area method, on average, uses only 6% more drivers, 7% longer total duration and 5% more miles than the lower bound. This shows that our core area method performs very well on the operational level. It provides a good balance for the tradeoff between the driver familiarity (hence consistency) and optimization flexibility.

6. CONCLUSIONS

This paper investigated the construction of routes for local delivery of packages in the presence of driver learning. To balance the tradeoffs of dispatch consistency and flexibility, we developed the concepts of "cell", "core area" and "flex-zone" for large-scale vehicle dispatching problems under stochastic demand, and explicitly considered the value of driver familiarity. Further, a two-stage vehicle routing model was developed: Strategic Core Area Design model and Operational Cell Routing model. The testing results showed that the tabu search meta-heuristic for the strategic model is capable of finding near-optimal solutions, as evidence by the lower bounds constructed for the nonlinear generalized assignment problem. Moreover, the core areas and flex-zone identified in the strategic level provides a good framework for building low cost yet consistent dispatch plans, as evidence by the favorable comparison to the no-core area method as well as by the lower bound constructed in the operational level for the learning-incorporated vehicle routing problems.

In our research, we assume that all customers have the same time commitment or time window. In reality, multiple time windows are more common. Therefore, future research will focus on developing more complicated core area design methods for multiple time windows. Work is also proceeding on implementing the methodology at UPS for routing its delivery vehicles.

Acknowledgement This research was supported by the National Science Foundation. We also acknowledge United Parcel Service for their support and participation in this project and the referees for their helpful comments in improving the paper.

REFERENCES

J. E. Beasley (1984). Fixed Routes. *J. Opl. Res. Soc.*, 35, 49-55.

J. E. Beasley and N. Christofides (1997). Vehicle Routing with A Sparse Feasibility Graph. *European J. of Operational Research*, 98, 499-511.

W. C. Benton and M. D. Rossetti (1992). The Vehicle Scheduling Problem with Intermittent Customer Demands. *Computers Ops Res.*, 19, 521-531.

D. Bertsimas (1992). A Vehicle Routing Problem with Stochastic Demand. *Operations Research*, 40, 574-585.

N. Christofides (1971). Fixed Routes and Areas for Delivery Operations, *International Journal of Physical Distribution*, 87-93.

G. Clarke and J. Wright (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points. *Operations Research*, 12, 568-581.

A. Campbell, L. Clarke, and M.W.P. Savelsbergh (2002). Inventory Routing in Practice: The Vehicle Routing Problem, *SIAM Monographs on Discrete Mathematics and Applications*, (P. Toth, D. Viego, Eds.), 309-330.

C. F. Daganzo (1984). The distance Traveled to Visit N Points with A Maximum of C stops Per Vehicle: An Analytical Model and An Application. *Tranport. Sci*., 18, 331-350.

S. E. Elmaghraby (1990). Economic Manufacturing Quantities under Conditions of Learning and Forgetting (EMQ/LaF). *Prod. Planning Control*, 1, 196-208.

A. Erera (2000). Design of Large-scale Logistics Systems for Uncertain Environments, Ph.D. Dissertation, University of California, Berkeley.

M. L. Fisher and R. Jaikumar (1981). A Generalized Assignment Heuristic for Vehicle Routing. *Networks*, 11, 109-124.

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

F. Glover and M. (1997). *Tabu Search*. Kluwer Academic Publishers, Netherlands.

R. W. Hall, Y. Du, and J. Lin (1994). Use of Continuous Approximations within Discrete Algorithms for Routing Vehicles: Experimental Results and Interpretation. *Networks*, 24, 43-56.

R. W. Hall (1996). Pickup and Delivery Systems for Overnight Carriers. *Transpn. Res. A*., 30, 173-187.

W. Hancock and F. Bayha (1992*). The Learning Curve*, in Handbook of Industrial Engineering, 2nd ed., Salvendy, G. (ed.), John Wiley.

M. Haughton (1998). The Performance of Route Modification and Demand Stabilization Strategies in Stochastic Vehicle Routing. *Transpn Res.-B*, 32, 551-566.

M. Haughton, A.Stenger (1998). Modeling the Customer Service Performance of Fixed-Rotes Delivery Systems under Stochastic Demands*. Journal of Business Logistics*, 9, 155-172.

M. Haughton (2000). Quantifying the Benefits of Route Reoptimization under Stochastic Customer Demands*,* *Journal of Operational Research Society*, 51, 320-322.

M. Haughton (2004). An Analysis of the Assignment of Delivery Routes to Vehicle Drivers in Stochastic Demand*,* 2nd World Conference on Production and Operations Management, Cancun, Mexico.

H. W. Kuhn (1955). The Hungarian Method for the Assignment Problem*, Naval Research Logistics Quarterly*, 2, 83-97.

M. Laguna, J.P Kelly, Gonzalez-Velarde and F. Glover (1995). Tabu Search for the Multilevel Generalized Assignment Problem. *European J. of Operational Research*, 82, 176-189.

G. Laporte (1992). The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms. *European J. of Operational Research*, 59, 345-358.

S. Lin (1966). Computer Solutions of the Traveling Salesman Problem. *Bell System Technical Journal*, 44, 2245-2269

G. F. Newell and C. F. Daganzo (1986). Design of Multiple Vehicle Delivery Vehicle Routes – I.A Ring Radial Network. *Transport. Res*., 20B, 345-364.

J. Potvin and J. Rousseau (1993). A Parallel Route Building Algorithm for the Vehicle Routing and Scheduling Problem with Time Windows. *European J. of Operational Research*, 66, 331-340.

M. Savelsbergh and M. Goetschalckx (1995). A Comparison of the Efficiency of Fixed versus Variable Vehicle Routes. *J. of Business Logistics*, 16, 163-187.

M. Solomon (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints. *Operations Research*, 35, 254-265.

W. R. Stewart and B. L. Golden (1983). Stochastic Vehicle Routing: A Comprehensive Approach. *European Journal of Operational Research,* 14, 371-385.

D. J. Waters (1989). Vehicle-scheduling Problems with Uncertainty and Omitted Customers. *J. Opl Res. Soc*., 40, 1099-1108.

K. F. Wong and J. E. Beasley (1984). Vehicle Routing Using Fixed Delivery Areas. *OMEGA Int. J. of Mgmt. Sci*., 12, 591-600.

T. Wright (1936). Factors Affecting the Cost of Airplanes. *Journal of Aeronautical Sciences*, 3, 122-128.

D. Zaret (1999). Vehicle Routing Algorithm for DPS. UPS Corporate IE Technical Report.

H. Zhong (2001). Territory Planning and Vehicle Dispatching with Stochastic Customers and Demand, Ph.D. Dissertation, University of Southern California.