Available online at www.sciencedirect.com
SCIENCE DIRECT®
Omega 33 (2005) 33–45
omega
The International Journal of Management Science
www.elsevier.com/locate/dsw

Dynamic vehicle routing for online B2C delivery

Timon C. Du[a], Eldon Y. Li[b,*], Defrose Chou[c]

[a] Department of Decision Sciences and Managerial Economics, The Chinese University of Hong Kong, Hong Kong
[b] Department of Information Management, College of Informatics, Yuan Ze University, Chung Li 320, Taiwan
[c] Department of Industrial Engineering, Chung Yuan Christian University, Taiwan

Received 19 June 2003; accepted 16 March 2004

Abstract

Electronic commerce (EC) is increasingly popular in today's businesses. The business-to-consumer EC environment has voluminous, unpredictable, and dynamically changing customer orders. A major part of the delivery system of this environment is the dynamic vehicle routing (DVR) system. This study investigates several algorithms suitable for solving the DVR problem in business-to-consumer (B2C) EC environment. It designs the solution process into three phases: initial-routes formation, inter-routes improvement, and intra-route improvement. A computer program is created to demonstrate a system simulating vehicle routing process under the online B2C environment. The simulated system collects data for system performance indexes such as simulation time, travel distance, delivery time, and delay time. The results show that when orders are placed through the Internet in an online B2C environment, the Nearest algorithms can be used to find satisfactory routes during the first phase of a DVR delivery system. The three-phase solution process is proven to be significantly better in travel distance and delivery time than the conventional single-phase solution process.

Keywords: Vehicle routing; Transportation; Algorithm; Route formation; System simulation; Electronic commerce; B2C; JIT delivery

1. Introduction

Delivering goods to customers is a critical activity in any business. On-time delivery relies heavily on effective vehicle routing once the merchandise is out the supplier's door and on its way to the customer. The problem of vehicle routing is much more complicated in an electronic commerce (EC) environment where the process of buying, selling, or exchanging products, services, and information is done through the Internet. In business-to-business (B2B) EC environment, the buyers and sellers are all business units. The main concern is how to maintain the efficiency of the supply chain partnership, which coordinates the order generation, the order taking, and the order fulfilment and distribution. In this environment, the buyers purchase products and services from the sellers with or without an intermediary. The two business partners integrate Just-in-Time (JIT) manufacturing and JIT inventory policy with JIT delivery. In fact, a JIT delivery service could be provided by either a buyer's, a seller's, or a third-party's deliverer (such as FedEx, UPS, or DHL). It is a coordinated effort of the deliverer and the seller or the suppliers of the seller. It is normal that these business partners have long-term relationships. The orders are normally planned, repeated, and reliable.

Contrary to B2B environment, the delivery policy of business-to-consumer (B2C) EC environment is different. The orders in online B2C environment are small in size, instantaneous, ever changing, and placed by numerous consumers. These customers are normally bargain seekers and care less about loyalty to the sellers. The coordination between the buyers and the sellers for JIT delivery is extremely difficult, if not impossible. When a customer places an order through the Internet, the best practice for the seller

* Corresponding author. Present address: California Polytechnic State University, College of Business, San Luis Obispo, CA 93407, United States. Tel.: +1 805 756 2964; fax: +1 805 756 1473.
E-mail address: eli@calpoly.edu (E.Y. Li).

0305-0483/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.omega.2004.03.005

is to ship the goods from an adjacent distribution center or partnering supplier. However, the availability of on-hand inventory (or safety stock) is very limited under JIT production setting and the goods will very likely be shipped from a distance depot. Therefore, in the B2C environment the need of having a quick-response vehicle dispatching system that handles dynamic demands of consumers is much greater than that in the B2B. This calls for an effective routing of delivery vehicles in order to minimize the travel distance and the delivery time.

There are various vehicle routing algorithms in the literature. However, none of them alone is useful in the online B2C environment. These algorithms work best only when customer orders are planned and can be predicted by the delivery system. This study adopts the heuristic approach of the existing dynamic vehicle routing technique to solve the delivery problem in the online B2C environment. The solution process is divided into three phases. The purposes of this study are: (1) to demonstrate a system simulating vehicle routing process under the online B2C environment, (2) to verify that the three-phase solution process performs significantly better than the single-phase solution process, and (3) to identify the optimal algorithms and improvement strategies for vehicle routing in the online B2C environment. The remaining paper is organized as follows. Section 2 reviews the existing literature on vehicle routing. Section 3 describes a three-stage dynamic vehicle routing process. Section 4 presents the architecture of the prototype of a dynamic vehicle routing system. Section 5 demonstrates the prototype system by running a simulation experiment. During the simulation, several route improvement strategies are tested using combinations of different existing algorithms. The performance indexes such as travel distance, service time, system time, and delay time are collected to analyze the simulation scenarios. Based on the simulation results, conclusions and recommendations about the delivery performance of these route improvement strategies are presented in Section 6.

2. Literature review

A review of literature reveals that the existing research into vehicle routing [1] or traveling salesman [2] problem focuses invariably on the JIT delivery in the B2B environment where orders are normally planned. There is a lacking of research for the online B2C environment because the coordination between buyers (i.e., consumers) and sellers for JIT delivery has been deemed neither feasible nor possible in the online B2C environment. Although we agree that the JIT inventory strategy popular in the online B2B environment is infeasible for the online B2C environment, it is our contention that the delivery performance could be improved in the online B2C environment by means of combining different vehicle routing algorithms available today. A review of relevant literature on vehicle routing problem and algorithms is presented in Sections 2 and 3.

2.1. Vehicle routing system

The solutions of vehicle routing problem (VRP) find the vehicle routes with the lowest cost given a known route map $G=(V,A)$, where $V=\{1,\ldots,n\}$ are the nodes of customers and the depot, and $A=\{(n_i,n_j) : n_i,n_j \in N, i \neq j\}$ are the links between the nodes [3]. Normally, there are three restrictions on solving the VRP: (1) each node, except the depot, can only be served once by only one vehicle; (2) all vehicles return to the depot after completing the service; and (3) all constraints have to be met. The examples of common constraints are vehicle capacity, the maximum visiting allowance, the total travel time, and the delivery time window. Using a formal mathematical expression, the VRP may be defined as [4]

$$\text{Minimize} \sum_{ijk} c_{ij}x_{ijk} \tag{1}$$

Subject to

$$\sum_{i} y_{ik} = \begin{cases} 1, & i=2,\ldots,n, \\ m, & i=1, \end{cases} \tag{2}$$

$$\sum_{i,k} q_i y_{ik} \leqslant Q_k, \quad i=1,\ldots,n,\ k=1,\ldots,m, \tag{3}$$

$$\sum_{i} x_{ijk} = \sum_{i} x_{jik} = y_{ik}, \quad i=1,\ldots,n,\ k=1,\ldots,m, \tag{4}$$

$$\sum_{i,j\in S} x_{ijk} \leqslant |S|-1, \quad \forall S \subseteq \{2,\ldots,n\}, \quad k=1,\ldots,m, \tag{5}$$

$$y_{ik} \in \{0,1\}, \quad i=1,\ldots,n,\ k=1,\ldots,m, \tag{6}$$

$$x_{ijk} \in \{0,1\}, \quad i,j=1,\ldots,n,\ k=1,\ldots,m, \tag{7}$$

where y_{ik} stands for node i served by vehicle k; c_{ij} is the delivery cost from nodes i to j; q_i is the ordering quantity of node i; Q_k is the capacity of vehicle k; $x_{ijk}=1$ means vehicle k drives from nodes i to j, m is the number of vehicles, and n is the number of nodes. Eq. (1) is the objective function to minimize the total traveling cost. The constraints of Eqs. (2)–(7) place a limit on the vehicle capacity and the travel flow.

Generally speaking, there are seven kinds of approaches to solving the VRP [5,6]:

(1) *Cluster-first route-second* approach divides the orders into several clusters and finds the most economic routes to ensure that the order deliveries are not assigned to inefficient routes. An example of this approach is the Sweep algorithm [7].
(2) *Route-first cluster-second* approach generates a vehicle route through all customers, and then divides the

route into several segments based on vehicle capacities. An example of this approach is the Space-Filling Curves algorithm [8].

(3) *Savings and insertion* approach assigns one vehicle to one order at first, and then merges the vehicles if the cost can be saved. It is a simple and efficient approach. The Insertion algorithm [9] used in this study belongs to this category.

(4) *Improvement and exchange* approach uses a heuristic approach to search for a better solution iteratively since solving the travel salesman problem (TSP) and VRP takes polynomial time. A famous example is K-OPT Exchange that replaces K original links by new links if the total cost is decreased. Note that the K-OPT is the extension of 2-OPT heuristic for the TSP from Lin [10] and 3-OPT from Lin and Kernighan [11].

(5) *Mathematical programming* approach finds the optimal solution in Relaxed Formulation [12].

(6) *Interactive optimization* approach relies on the knowledge and experience of decision-makers and revises the system parameters through an interactive interface.

(7) *Exact procedures* approach can find optimal solutions for relatively smaller problems using algorithms. Examples are the Branch and Bound method and Dynamic Programming.

Many different constraints have been considered in VRP and evolve the problem into different forms. For example, if the delivering time is considered, the problem becomes a VRP with time window. If backhauling is allowed, the problem is a VRP with backhauling. The time window problems can be further differentiated into a soft time window or a hard time window, in which the soft time window means the delivery time can be missed if one pays for the penalty. In contrast, the late delivery service will be rejected in the hard time window condition. For the VRP, the objective is to minimize the total cost rather than simply the distance.

2.2. Dynamic VRP

The existing vehicle routing (VR) algorithms are best for solving VR problems where orders are planned and repeated. However, in today's information age, many orders are placed through the Internet. This makes the VR problem become more dynamic, thus the dynamic vehicle routing (DVR) problem has drawn more attention. The DVR algorithm allows vehicles to update services based on renewed information [13]. The major difference between the VR and the DVR algorithm is that the input data of the VR are not changed, i.e., the parameters are predetermined, while those of the DVR are uncertain [14]. For example, the consumers' orders and the vehicle travel time are changed from time to time in the DVR problem (DVRP). These input data are known only moment before determining the vehicle routes. Therefore, this vehicle routing problem is also called the dynamic, the real-time, or the on-line problem [15]. Also, the DVRP is different from the dynamic assignment problem (DAP) [14] in the sense that the DVRP serves a sequence of consumers while the DAP serves one consumer at a time. Powell finds that solutions for the dynamic scheduling and routing problem (DSRP) can adapt algorithms from static VRPs [13]. The approaches are normally conducted by either (1) re-solving the routing plan either locally (e.g., by the Insertion Algorithm) or globally (e.g., by the Tabu Search) when the input data is updated [16]; or (2) by using Markov Decision Process [17] or Stochastic Programming [13] to re-optimize the problems. However, when a problem grows, the conventional approaches are not efficient enough. To provide efficient results, some studies use neural networks [19] while others use parallel Tabu Searchs [20].

Two main types of approaches have been adopted to solve DVRPs [2,3,16]:

(1) Approaches that have been adapted from static VRPs, such as re-optimization approaches and heuristic approaches. The former type re-optimizes the solution whenever a new event occurs, while the latter re-organizes local solutions in response to changing events. Examples of re-optimization approaches can be found in the work of Dial [21]. There is abundant literature on using heuristic approaches to solve static VRP-related problems. Some examples for local improvement are the Insertion/Saving Algorithm [2,3], the K-Opt and Or-Opt Algorithms [22–24] and the λ-Exchange Algorithm [25]. These algorithms can be used either for generating initial solutions or for improving route assignments. Moreover, some meta-heuristic algorithms are used for overall improvement: for example, the Tabu Search Algorithm [26,27] and the Genetic Algorithm [23,24,28]. Due to the emerging needs, heuristic approaches are also applied to solving DVRPs.

(2) Stochastic methods, such as the Markov Decision Process [17] and Stochastic Programming [13]. However, these approaches are limited in their ability to handle large-scale problems. Some heuristic algorithms that belong to this category can also be used, such as the Simulated Annealing Algorithm [25] and the Noising Method [29]. However, these approaches are also limited by the problem of high computation time.

It is noteworthy that the VRP heuristic approach was derived from the TSP, which solved the problem of a single delivery carrier [2]. This heuristic approach is normally composed of three phases.

(1) *Initial-routes formation*. This phase generates initial vehicle routes based on the matrix-like order distances. Examples include Insertion algorithms, such as the Nearest Neighbor algorithm [30,31] and the Savings algorithm [9], the Sweep procedure [32], and the Minimum Spanning Tree algorithm [33].

(2) *Routes improvement*. The second phase improves the initial routes proposal by exchanging or inserting nodes or links. If such exchanging or inserting is done between the routes, it is called "inter-route improvement." In contrast, if it is within the route, it is called "intra-route improvement." The algorithms can be used for both inter-routes improvement and intra-route improvement are K-OPT and or-OPT [2].

(3) *Fine tuning*. This phase revises the proposal from inter-routes improvement by using generalized heuristic algorithms, such as the Tabu Search algorithm [26,34], the Simulated Annealing algorithm [35], the Genetic algorithm [23,24], and the Threshold Values Acceptance method [2].

3. Dynamic VR for JIT delivery

Since consumers can place orders through the Internet under online B2C environment, the orders are unpredictable, but the service response is expected to be fast. One way to meet the consumers' needs is to find an algorithm that can satisfy both the capacity and time-window constraints in a timely manner. In order to find feasible solutions, this study adopts the soft-time-window policy, but the severity of the late penalty is not taken into account.

Note that the solution to the DVRP is combinatorial [36,37]. According to the study of Bodin and Golden [5], this type of problem can be solved by a staged approach such as the cluster-first route-second method. This study adopts a mixed strategy that is composed of three phases: initial-routes formation, inter-routes improvement, and intra-route improvement.

3.1. Initial-routes formation

To find the initial routes, orders are first divided into clusters and the vehicle delivery route is formed in each cluster. Then, a new order is placed into a specific location and the routes are rescheduled. The re-scheduling process has to be punctual and flexible. This study adopts four algorithms for the initial-routes formation: First-In-First-Serve, First-Fit-Nearest, Best-Fit-Nearest, and the Sweep procedure. First-In-First-Serve is self-explanatory. It simply serves the orders based on the sequence of the orders being placed. Other algorithms are illustrated below, beginning with the fundamental algorithm, the Insertion/Saving algorithm.

3.1.1. The insertion/saving algorithm with time window

This algorithm, proposed by Clarke and Wright [9], is the most commonly used approach among the Insertion algorithms. The algorithm solves the VRP problem with unlimited vehicles but limited capacities. The algorithm links nodes to the depot individually and merges the links with the saving, which is calculated as the difference between the new route and the original route. For example, assuming two nodes (orders) i and j are served by two vehicles, if the service is offered by one vehicle, the saving is

$$S(i,j) = 2d(1,i) + 2d(1,j) - [d(1,i) + d(i,j) + d(j,1)]$$
$$= d(1,i) + d(1,j) - d(i,j), \quad (8)$$

where $S(i,j)$ is the saving, $d(i,j)$ is the distance between the two nodes, and node 1 stands for the depot. The Insertion/Saving algorithm normally ignores the fixed expense of vehicles and only considers the total distance [3]. Since this algorithm is limited to solve problems with constraints, it is more appropriate to use it for obtaining initial solutions.

3.1.2. The Nearest algorithm with time window

The Nearest algorithm uses the Insertion/Saving approach to assign new orders in two ways: First-Fit-Nearest and Best-Fit-Nearest [38]. The First-Fit-Nearest algorithm assigns the new order to the last stop of the first feasible solution, while the Best-Fit-Nearest algorithm puts the new order in the most economic route after examining all the feasible solutions.

The implementation steps of the First-Fit-Nearest algorithm are: (1) When a consumer places a new order, beginning with vehicle $i = 1$, take the following action. (2) Evaluate the capacities of all vehicles to find the available vehicle. If the capacities of all vehicles are full, stop; otherwise do as follows. (3) Calculate the distance from the last node of the available vehicles to the new order. (4) Check the time-window constraint of the available vehicle with the lowest cost. If the time window can be met, go to the next step; otherwise, choose the vehicle with the second lowest cost. If none of the vehicles can deliver the order on time, the vehicle with the minimum delay should be chosen. (5) Insert the new order as the last node in the chosen vehicle.

The Best-Fit-Nearest algorithm is similar to the First-Fit-Nearest algorithm, except steps (3) and (5): (1) When a consumer places a new order, beginning with vehicle $i = 1$, take the following action. (2) Evaluate the capacity of all vehicles to find the available vehicle. If the capacities are all full, stop; otherwise, do as follows. (3) Calculate the distance from all the nodes of the available vehicles to the new order, and find the insertion point with the lowest cost. (4) Check the time-window constraint of the vehicle with the lowest cost. If the time window can be met, go to the next step; otherwise, choose the vehicle with the second lowest cost. If none of the vehicles can deliver the order on time, the vehicle with the minimum delay will be chosen. (5) Insert the new order to the best position in the sequence of the chosen vehicle.

3.1.3. The Sweep procedure with time window

The Sweep procedure was initially proposed by Wren and Holliday [7] in 1972, and was named by Gillett and Miller [32] in 1974. The Sweep procedure divides the orders into regions and assigns one vehicle to deliver orders in a region.

The new order belongs to the closest vehicle. The algorithm is as follows: (1) Starting from the depot, the service area is divided by the number of vehicles with equal degrees of angles in the polar coordinate. (2) When a consumer places a new order, find the region the order belongs to. (3) Evaluate the capacity of the vehicle serving that region. If the capacity is full, go to step 4; otherwise, go to step 6. (4) Check if the depot has available vehicles and assign the vehicle to the new order, then go to step 6; if no vehicle is available, go to step 5. (5) Assign the order to the vehicles in closer regions if they are available. If no vehicle is available, stop; otherwise, do as follows. (6) Calculate the distance from all nodes of the available vehicles to the new order, and find the insertion point with the lowest cost. (7) Check the time-window constraint of the vehicle with the lowest cost. If the time window can be met, go to the next step; otherwise, go back to step 5. If none of the vehicles can deliver the order on time, the vehicle with the minimum delay should be chosen. (8) Insert the new order into the best position in the sequence of the chosen vehicle.

3.2. Inter-routes improvement

One can switch either a node or a link of a route with another route to improve routing efficiency. This study uses the Insertion algorithm to demonstrate the switch of nodes and the 2-Exchange algorithm [39] for the switch of links between routes. The description of these two algorithms follows.

3.2.1. The Insertion algorithm with time window

The Insertion algorithm removes one node from a route and adds it to the other route to shorten the total travel distance. The removed node must be an unselected order and be inserted into another route. The algorithm is: (1) Select a route with one order that has not been chosen, and then carry out the following action. (2) Remove the order node and insert it into the other route. (3) If the total travel distance is smaller than the original distance, go to step 4; otherwise, reverse the node back to its original position and repeat from step 1. Stop the procedure if no improvement can be made. (4) Evaluate the capacity of the vehicle on the route being inserted with the removed node. If the capacity is full, reverse the node back to its original position and repeat from step 1; otherwise, go to step 5. (5) Check the time-window constraint of both vehicles to see whether or not the total delay time is smaller than the original setup. If yes, go to the next step; otherwise, reverse the node back to its original position and repeat from step 1. (6) Confirm the new routes and repeat from step 1.

3.2.2. The 2-Exchange algorithm with time window

The 2-Exchange algorithm cuts two routes into four segments, swaps the tail segments, and then reconnects them. The algorithm can be implemented as follows: (1) Select two routes with orders that have not been chosen. Cut the links before the orders of both routes and do as follows. (2) Reconnect the tail of the two segments to the other route. Reconnect the other tail segment to the opposite route. (3) If the total travel distance is smaller than the original distance, go to step 4; otherwise, reverse the routes back to their original flows and repeat from step 1. Stop the exchange if no improvement can be made. (4) Evaluate the capacity of both vehicles. If either capacity is full, reverse the routes back to their original flows and repeat from step 1; otherwise, go to step 5. (5) Check the time-window constraint of the two vehicles to see whether or not the total delay time is smaller than the original total delay time. If yes, go to the next step; otherwise, reverse the routes back to their original flows and repeat from step 1. (6) Confirm the new routes and repeat from step 1.

3.3. Intra-route improvement

Two algorithms are chosen to improve the performance within a route: or-OPT and 2-Swap. These algorithms are chosen because they are simple and effective. Also, Potvin et al., and Potvin and Bengio [23,24] had shown that or-OPT is efficient at solving the VRP with a time-window constraint. The study of Solomon et al. [40] has also shown that or-OPT performs better than 3-OPT in the VRP with a time-window constraint. Similarly, the 2-Swap algorithm normally produces solutions with shorter distance by serving closer nodes together, thus is chosen as the improvement algorithm.

3.3.1. The or-OPT algorithm with time window

The or-OPT algorithm removes a node from a route and relocates the node to another position of the route if the total travel distance can be shortened. The algorithm is as follows: (1) Choose an unselected node from a route that has not been processed and then do as follows. (2) Insert the node into a position other than the original position. (3) Find the position with the minimum total distance, then go to step 4; if it cannot be found, reverse the node back to its original position and repeat from step 1. Stop when no more improvements can be made. (4) Check the time-window constraint of the vehicle to see whether or not the total delay time is smaller than the original one. If yes, go to the next step; otherwise, reverse the node back to its original position and repeat from step 1. (5) Confirm the new sequence of the route and repeat from step 1.

3.3.2. The 2-Swap algorithm with time window

The 2-Swap algorithm chooses any two nodes on a route and swaps the nodes if the total travel distance can be shortened. The algorithm is as follows: (1) Choose any two unselected nodes from a route and do as follows. (2) Swap these two nodes and calculate the new travel distance. (3) Find a pair of nodes with the minimum distance, then go to step 4; if such nodes cannot be found, reverse the nodes back to their original positions and repeat from step 1. Stop when no more improvements can be made. (4) Check the

time-window constraint of the vehicle to see whether or not the total delay time is smaller than the original one. If yes, go to step 5; otherwise, reverse the nodes back to their original positions and repeat from step 1. (5) Confirm the new sequence of the route and repeat from step 1.

4. System design and development

A dynamic VR system is developed to simulate the system performance in online B2C environment. The system has five subsystems: the *control subsystem*, the *vehicle subsystem*, the *route list subsystem*, the *customer subsystem*, and the *algorithm subsystem*. Also, one system function, *simulation clock*, is used to set timepieces of simulation for performance measurement. Fig. 1 represents the architecture in the object modeling technique (OMT) model [41].

(1) *The Control Subsystem*. The control subsystem manages the operations among four other subsystems. The operations include placing consumers' orders, deciding service vehicles, choosing the algorithm, proceeding with route improvement, and controlling the system clock. Other system functions are the system clock, random number generators, event tables, and service records. The major system events are *start*, *end*, *wait*, *return*, *service*, *travel*, *order*, and *rest*.

(2) *The Vehicle Subsystem*. The vehicle subsystem includes five operations: (a) set initial status, (b) set all vehicles as *idle* if no orders are in the customer list, (c) set vehicle as *serve* while proceeding with the delivery, (d) set vehicle as *travel* after finishing the servicing and check the customer list for the next service, and (e) set vehicle as *return* if the delivery process is completed. Also, functions such as orders addition/deletion, performance evaluation, customer list creation, and vehicle routes generation are provided by this subsystem.

(3) *The Route List Subsystem*. Each vehicle has its own itinerary. The itinerary records information such as service sequence and types of services. The travel distance is calculated by this function.

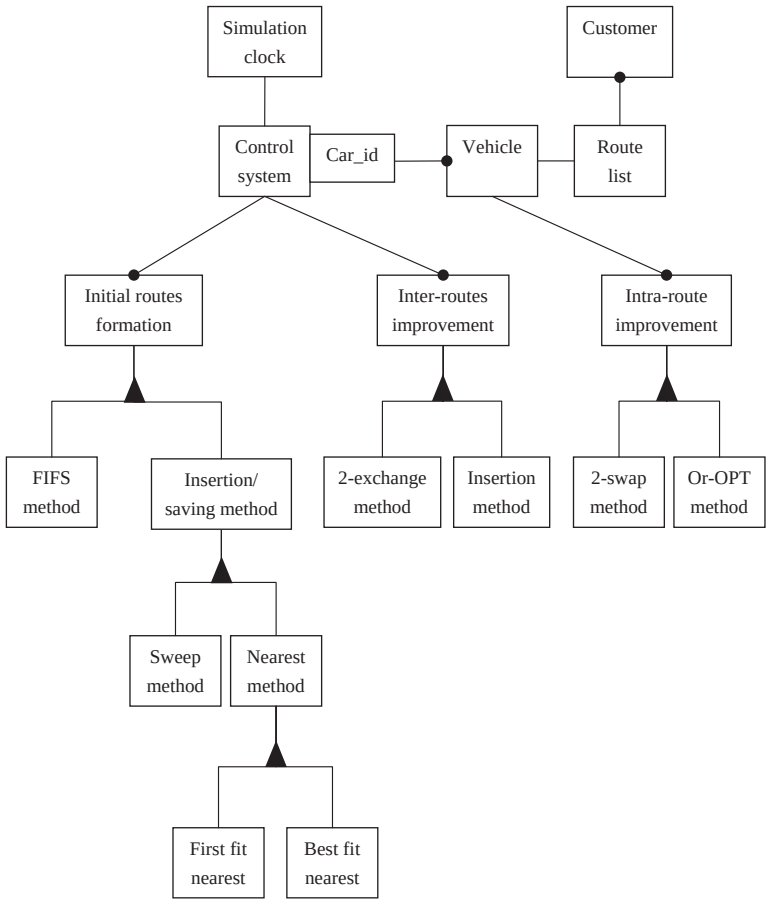

Fig. 1. The DVR system.

(4) *The Customer List Subsystem*. The customer list records all the information about customers. Each vehicle has a customer list recording the serviced customers and the completed time.
(5) *The Algorithm Subsystem*. This subsystem provides the algorithms for initial-routes formation, inter-routes improvement, and intra-route improvement.

The consumer orders (i.e., order locations, order quantities, and order time) in this study are simulated by random number generators using different probability distributions. When the simulation begins, the vehicles have the status of *start*. Then, the simulator uses the initial-routes formation algorithm to assign orders to vehicles when orders are placed. The vehicle *travels* to the order location and *serves* the consumer. If no orders remain, the vehicle *waits*; otherwise, the vehicle continues *traveling*.

The system is implemented under the following schemes. The journey of the vehicles starts from the depot, which is located at coordinate $(50, 50)$. The order arrivals are generated by the Poisson distribution with a mean arrival rate of 30 orders per 60 time-units. The delivery locations are randomly generated in the square of $(0, 0)$ to $(100, 100)$ and the vehicles travel along the rectangular coordinate system. That is, the largest distance between the depot and a delivery location is 100 and the largest distance is 50 in each quadrant. The service time for each order at the customer location is set to 10 time-units. The travel time per unit of distance is one time-unit. The quantity of an order is randomly generated between 10 and 50, i.e., the average quantity per order is 30. The system will stop when the total number of orders delivered reaches 100. Therefore, the total quantities need to be delivered is around 3000. The system assumes each vehicle has a quantity capacity of 200, thus 15 vehicles are utilized in the system. The number of vehicles, 15, is fixed in this experiment because our attention is focused on the performance of different algorithms instead of optimizing the utilization of vehicles. When a new order is placed, the system assigns the order to a vehicle according to the algorithms of the initial-routes formation, the inter-routes improvement, and the intra-route improvement. If the remaining order quantity of a vehicle is less than 25, the vehicle returns to the depot to replenish the order quantity. The simulation clock is set to zero at the moment the simulation begins, and all vehicles are set to the status *start*. The vehicles are set to the status *wait* if the services are completed. The earliest service starting time of the any time window is the system start time, i.e., time zero. The service completion time requested by a customer is generated between 20 and 240. For example, an order may be placed at time 5 and the buyer requests the delivery service to be completed by time 100; therefore, the time window of this order is $(5, 100)$. Both time values of 5 and 100 are generated from the random number generator of Visual C++. When an order is placed, the system will assign the order to one vehicle based on one of the initial-routes formation algorithms. Then, the inter-routes improvement algorithm and/or the intra-route improvement algorithm are activated, depending on our experimental settings (see next section). The simulation program is written in Visual C++ 6.0 and implemented on a PC platform. Minitab is used for analysis of variance (ANOVA) at a confidence level of 99%.

5. Experimental design and results

In order to examine the performance of different DVR improvement strategies, five performance indicators are collected: (1) total simulation time, (2) averaged vehicle travel distance, (3) averaged delivery time, (4) averaged delay time, and (5) averaged delivery time for new orders. The simulation time records the time when all the orders are served. Regardless of the various experimental settings, the smaller the total simulation time the better the DVR system performance. The smaller averaged vehicle travel distance stands for running the delivery system at a lower cost. The delivery time is the elapsed time from the order is placed to the time the delivery is in the hands of the consumer. The delay time means that the delivery cannot be completed before the ending time window of an order. Since this study adopts the soft-time-window strategy, the delay time is obtained from deducting the actual delivery completion time by the ending time window. Alternately, the delay time may be calculated by subtracting the width of the consumer's time window from the delivery time if the latter is longer. That means, the smaller the delay time, the better the algorithm's performance in the online B2C delivery system. Therefore, the satisfaction of consumers can be observed from both the averaged delivery time and the averaged delay time.

The experiment has two initial conditions: 0 or 50 consumer orders in system before algorithms start running. The system will stop when the total number of orders delivered reaches 100. The system experiments with the combinations of three groups of algorithms, i.e., initial-routes formation algorithms, inter-routes improvement algorithms, and intra-route improvement algorithms. Both inter-routes improvement and intra-route improvement have three settings, i.e., no improvement, 2-Exchange algorithm, and Insertion algorithm for inter-routes improvement; and no improvement, or-OPT algorithm, and 2-Swap algorithm for intra-route improvement.

Moreover, a uniform distribution is used to generate three different sets of order locations for the consumers and each set of location is considered as a blocking factor in our design of experiment. Therefore, the experiment has two initial order settings, four initial-routes formation algorithms, three levels of inter-routes improvement, three levels of intra-route improvement, and three sets of delivery locations, calling for $216 (= 2 \times 4 \times 3 \times 3 \times 3)$ simulation runs in total. For each of the two initial order settings, there are 108 simulation runs. Table 1 presents the results of simulation having no initial orders, using four initial-routes formation

Table 1
The simulation results without initial orders

Initial route formation	Model	Total simulation time	Averaged vehicle travel distance[a]	Averaged delivery time[b]	Averaged delay time[b]	New-order averaged delivery time[b]
First-In-First-Serve	1	851.5	347.0	257.7	200.9	257.7
	2	591.1	220.4	173.5	117.5	173.5
	3	635.4	244.6	188.5	132.0	188.5
	4	516.2	188.8	153.7	99.1	153.7
	5	455.7	166.7	143.8	89.3	143.8
	6	477.2	170.3	140.1	86.3	140.1
	7	567.3	232.4	178.3	122.3	178.3
	8	520.1	199.4	156.4	100.7	156.4
	9	515.1	205.3	160.0	102.5	160.0
First-Fit-Nearest	1	289.3	142.6	77.3	30.7	77.3
	2	287.8	124.3	64.9	23.3	64.9
	3	287.8	125.7	65.5	23.4	65.5
	4	284.8	133.1	70.4	26.0	70.4
	5	265.1	124.3	64.4	23.1	64.4
	6	287.1	124.0	65.4	23.3	65.4
	7	274.5	134.5	71.8	26.1	71.8
	8	274.9	120.6	63.2	22.5	63.2
	9	274.2	122.8	63.8	22.7	63.8
Best-Fit-Nearest	1	357.8	131.4	78.0	34.4	78.0
	2	321.2	115.5	74.7	32.0	74.7
	3	322.8	115.8	75.2	32.3	75.2
	4	350.4	114.4	75.7	31.7	75.7
	5	361.3	106.0	74.2	30.9	74.2
	6	358.9	111.8	76.8	34.3	76.8
	7	378.5	123.7	74.8	32.5	74.8
	8	341.8	116.7	74.5	32.6	74.5
	9	332.8	115.0	76.9	34.6	76.9
Sweep	1	460.8	176.1	108.2	61.0	108.2
	2	430.6	160.0	101.2	52.7	101.2
	3	430.6	160.0	102.1	52.6	102.1
	4	454.8	153.6	95.5	48.8	95.5
	5	406.5	140.9	91.7	44.0	91.7
	6	408.7	140.5	95.0	48.4	95.0
	7	411.2	158.2	99.3	52.4	99.3
	8	369.8	144.7	92.2	47.5	92.2
	9	382.6	149.4	94.8	47.5	94.8

[a]Total distance divided by 15 vehicles.
[b]Total time divided by 100 orders.

strategies and the following nine models:

(1) No inter-routes improvement; no intra-route improvement
(2) No inter-routes improvement; or-OPT intra-route improvement
(3) No inter-routes improvement; 2-Swap intra-route improvement
(4) Insertion inter-routes improvement; no intra-route improvement
(5) Insertion inter-routes improvement; or-OPT intra-route improvement
(6) Insertion inter-routes improvement; 2-Swap intra-route improvement
(7) 2-Exchange inter-routes improvement; no intra-route improvement
(8) 2-Exchange inter-routes improvement; or-OPT intra-route improvement
(9) 2-Exchange inter-routes improvement; 2-Swap intra-route improvement.

Furthermore, Table 2 shows the results with 50 initial orders in the system. A scrutiny of Tables 1 and 2 reveals that most values of total simulation time and averaged delay time

Table 2
The simulation results with 50 initial orders

Initial route formation	Model	Total simulation time	Averaged vehicle travel distance[a]	Averaged delivery time[b]	Averaged delay time[b]	New-order averaged delivery time[c]	Difference between initial and new orders
First-In-First-Serve	1	739.1	352.0	249.5	166.3	258.3	(17.6)[d]
	2	537.7	225.4	172.4	92.4	175.7	(6.6)
	3	568.8	247.2	188.9	107.7	190.9	(4.0)
	4	492.9	204.3	172.7	92.2	166.1	13.2
	5	450.3	195.7	145.3	69.8	136.3	18.0
	6	456.3	187.6	150.7	73.6	137.5	26.4
	7	574.2	229.1	185.5	102.1	194.7	(18.4)
	8	497.2	199.6	153.5	77.9	146.2	14.6
	9	472.8	214.1	162.5	83.9	162.7	(0.4)
First-Fit-Nearest	1	292.8	146.9	91.7	31.1	104.6	(25.8)
	2	279.3	124.1	78.3	21.0	70.1	16.4
	3	289.7	127.0	79.3	22.9	75.3	8.0
	4	278.4	125.4	81.0	24.8	80.0	2.0
	5	258.1	111.2	73.3	17.2	66.5	13.6
	6	265.1	110.3	72.9	18.1	67.5	10.8
	7	281.3	141.8	89.0	28.5	99.3	(20.6)
	8	280.4	125.0	78.2	20.7	70.5	15.4
	9	284.3	127.3	80.0	22.5	76.5	7.0
Best-Fit-Nearest	1	330.0	132.6	92.2	28.0	77.2	30.0
	2	362.5	120.8	92.0	28.4	86.2	11.6
	3	351.7	121.5	90.2	28.6	84.2	12.0
	4	353.9	130.9	91.6	28.3	84.2	14.8
	5	372.1	114.5	89.1	26.7	82.9	12.4
	6	323.4	120.1	86.0	26.3	77.9	16.2
	7	323.5	125.3	89.6	27.0	77.1	25.0
	8	332.3	122.0	90.4	26.6	81.8	17.2
	9	321.8	117.5	87.7	26.3	79.9	15.6
Sweep	1	411.5	172.9	114.1	47.2	108.5	11.2
	2	390.8	154.1	102.4	35.8	97.4	10.0
	3	369.2	156.9	102.4	36.1	96.5	11.8
	4	384.9	156.4	102.5	37.0	96.3	12.4
	5	398.9	151.6	103.0	37.8	101.5	3.0
	6	410.5	159.0	104.3	39.2	98.7	11.2
	7	380.8	154.8	104.0	39.2	96.1	15.8
	8	443.2	159.2	103.1	35.4	98.9	8.4
	9	427.8	155.1	104.4	36.8	99.1	10.6

[a]Total distance divided by 15 vehicles.
[b]Total time divided by 100 orders.
[c]Total time divided by 50 orders.
[d] $= 2\times$ (the averaged delay time of the 50 initial orders − the average delay time of the 50 new orders).

have improved by having initial orders, while most values of averaged vehicle travel distance, averaged delivery time, and new-order averaged delivery time exacerbated. A detailed discussion of simulation outcomes follows.

5.1. Simulation time

The ANOVA of total simulation time shows that the interaction of initial-routes formation, inter-routes improvement, and intra-route improvement is significant at $p < 0.01$. This indicates that all three sets of algorithms determine the length of simulation time. They cannot be considered independently of each other. As for the main factor of initial order setting (0 or 50 initial orders), it significantly affects the total simulation time at $p < 0.01$ level. The results show that when 50 orders are initially in the system, the shortest simulation time is 258.1 s. After plotting the interaction diagrams, the results show that combining the First-Fit-Nearest

algorithm and the Insertion algorithm (Models 4, 5, 6) can produce a shorter simulation time (from 258.1 to 287.1). Also, the First-Fit-Nearest algorithm can generate a lower simulation time with any algorithms for intra-route improvement (279.3–289.7). In contrast, the First-In-First-Serve algorithm has the longest simulation time when no algorithm is used for inter-routes improvement (739.1 and 851.5). A combination of the Insertion algorithm and the or-OPT algorithm (Model 5) produces the lowest simulation time (265.1 and 258.1).

5.2. Vehicle travel distance

The ANOVA of the averaged vehicle travel distance shows that the interactions of (1) initial order setting, initial-routes formation, and inter-routes improvement, and (2) initial-routes formation, inter-routes improvement, and intra-route improvement, are significant ($p < 0.01$). Therefore, the averaged travel distance is determined by all three factors. The results show that the single-phase solution process (Model 1) has much longer distance to travel than the two-phase (Models 2, 3, 4, 7) or three-phase (Models 5, 6, 8, 9) solution process. The averaged travel distance is shortest (106.0) when the system is initially empty and uses Best-Fit-Nearest formation, the Insertion algorithm, and the or-OPT algorithm (i.e., Model 2). Also, using or-OPT for intra-route improvement (Models 2, 5, 8) in general can generate a shorter travel distance (115.5, 106.0, 116.7) when the Best-Fit-Nearest algorithm is used for initial-routes formation and the initial number of orders is empty.

5.3. Averaged delivery time

The ANOVA of the averaged delivery time reveals that the interaction of initial-routes formation, inter-routes improvement, and intra-route improvement is significant ($p < 0.01$). Also, the main factor of initial order setting significantly affects the averaged delivery time at $p < 0.01$ level. By further examining the data, we found that the single-phase solution process (Model 1) has much longer averaged delivery time than the two-phase (Models 2, 3, 4, 7) or three-phase (Models 5, 6, 8, 9) solution process. The averaged delivery time is the lowest (63.2) when the system is without initial orders and uses the First-Fit-Nearest formation, the 2-Exchange algorithm, and the or-OPT algorithm (i.e., Model 8). In general, the 2-Exchange algorithm (Models 7–9) can produce a lower averaged delivery time when the system is initially empty, while the Insertion algorithm (Models 4, 5, 6) can produce a lower averaged delivery time when the system is initially filled with 50 orders.

5.4. Averaged delay time

The ANOVA of averaged delay time shows that the interactions are significant ($p < 0.01$) for (1) initial setting, initial-routes formation, and inter-routes improvement, and (2) initial-routes formation, inter-routes improvement, and intra-route improvement. The results show that when 50 orders are initially in the system and the Model 5 is used, the delay time is shortest (17.2). Also, the same set of algorithms that performs best in the averaged delivery time is the best combination for the averaged delay time. That is, the 2-Exchange algorithm (Models 7–9) can produce a lower averaged delay time when the system is initially empty, while the Insertion algorithm (Models 4, 5, 6) can produce a lower averaged delay time when the system is initially filled with 50 orders.

5.5. Averaged delivery time for new orders

Next, we consider the averaged delivery time for new consumer orders and its impact on existing consumers. From the simulation, it can be seen that the averaged delivery time for new orders is the same as the averaged delivery time for all orders if no initial orders exist in the system. When the system has 50 initial orders, the 15 vehicles are available to handle the deliveries. Any new order arriving after the system starts will be assigned to a vehicle, according to the algorithms. The simulation results show that the averaged delivery time for the new orders may be higher or lower, depending on the algorithms used. The last column of Table 2 shows the difference of averaged delivery time between the new 50 orders and the 50 initial orders. It shows most of the differences are positive, indicating the delivery time of new orders improves as more and more orders are delivered. The best combination for 50 initial orders is to use the First-Fit-Nearest algorithm for initial-routes formation, the Insertion algorithm for inter-routes improvement, and the or-OPT algorithm for the intra-route improvement (Model 5).

Table 3 shows the improvements from the worst combination to the best combination in either with or without initial orders in the system. The improvements are significant and range from 65% to 90% (Fig. 2).

6. Conclusions and recommendations

A computer program demonstrating the dynamic vehicle routing system in an online B2C environment was created to collect data such as simulation time, vehicle travel distance, delivery time, and delay time in this study. These indexes enable us to select the best combination of existing algorithms for routing vehicles efficiently and effectively under different conditions in the online B2C environment. The result of simulation reveals that if there are no orders in the system initially, the First-Fit-Nearest algorithm has the smallest simulation time when it is combined with Model 5. It can find the smallest vehicle travel distance, averaged delivery time, averaged delay time, and new-order averaged delivery time, when combined with Model 8.

Table 3
Comparison of improvement between the best combination and the worst combination

Performance index	Initial orders	Worst combination	Best combination	Improvement (%)
Total simulation	No	851.5	265.1	69
	Yes	739.1	258.1	65
Averaged vehicle travel distance	No	347	106	69
	Yes	352.0	110.3	69
Averaged delivery time	No	257.7	63.8	75
	Yes	249.5	72.9	71
Average delay time	No	200.9	22.5	89
	Yes	166.3	17.2	90
New-order averaged delivery time	No	257.7	63.8	75
	Yes	258.3	66.5	74

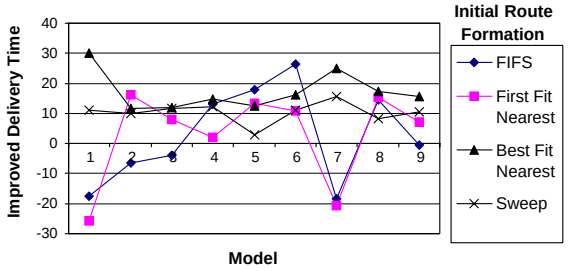

Fig. 2. The Delivery time change of existing orders caused by the new orders.

On the other hand, if the system does have initial orders, the best combination is the First-Fit-Nearest algorithm with Model 5 for finding the smallest simulation time, averaged delay time, and new-order averaged delivery time. Combining the First-Fit-Nearest algorithm with Model 6 yields the smallest vehicle travel distance and averaged delivery time. Regardless of whether or not the system has initial orders, the worst combination is the First-In-First-Serve algorithm with Model 1 and the single-phase solution process (Model 1) has much longer averaged travel distance and delivery time than the two-phase (Models 2, 3, 4, 7) or three-phase (Models 5, 6, 8, 9) solution process.

Furthermore, we can draw the following conclusions and recommendations.

(1) The algorithm that diminishes the impact of the delivery time caused by new orders cost (i.e., the smallest vehicle travel distance) is a combination of Best-Fit-Nearest for initial-routes formation, Insertion for inter-routes improvement, and or-OPT for intra-route improvement (Model 5).
(2) Customer satisfaction is highest (i.e., the smallest averaged delivery time and averaged delay time) if the First-Fit-Nearest algorithm is used for initial-routes formation together with either or-OPT or 2-Swap for intra-route improvement. In this case, the inter-routes improvement algorithm should use 2-Exchange when the system has no initial order or use Insertion when the system has initial orders.
(3) The best algorithm to perform initial-routes formation for new orders is First-Fit-Nearest or Best-Fit-Nearest. The data reveal that or-OPT, 2-Swap, Insertion, and 2-Exchange do not significantly improve the new orders.
(4) Regarding the satisfaction of both existing consumers and new consumers (the smallest averaged delivery time, averaged delay time, and new-order averaged delivery time), the best combination is using the First-Fit-Nearest or Best-Fit-Nearest algorithm for initial-routes formation; Insertion for inter-routes improvement; and either or-OPT or 2-Swap for intra-route improvement.

7. Managerial implications

Vehicle routing problem is a complex issue in online B2C environment in which orders are voluminous, unpredictable, and dynamically changing. There are various vehicle routing algorithms in the literature. However, none of them alone is useful in the online B2C environment. This study examines several combinations of existing algorithms suitable for solving the dynamic vehicle routing problem and investigates the strategies of combining different vehicle routing techniques for quick-response delivery in the online B2C environment. It adopts a heuristic approach of dynamic vehicle routing to solve the online B2C delivery problem in three phases.

In the online B2C environment where orders are placed through the Internet, one must have an efficient transportation system to support the order deliveries. We strongly encourage the use of the First-Fit-Nearest or the Best-Fit-Nearest algorithm for initial-routes formation. If the system needs to recommend routes promptly, these two algorithms can do so without the help of the intra-route improvement and the inter-routes improvement algorithms.

The conclusions obtained in this study can be used to improve the service quality in B2C. For example, when the waiting time between placing an order and receiving the order can be significantly decreased, various products and services (such as ordering a lunch box) can be sold via B2C service. Moreover, together with the newly developed technologies, such as radio frequency identification (RFID), the consumer can trace his/her order precisely.

Finally, this study only included a limit number of factors influencing the performance of a delivery system. There are other factors that could be considered in future studies. These include various vehicle capacities, multiple depots, a time-constraint penalty, both pickup and delivery services, other heuristic algorithms, road conditions, and order information. Furthermore, a supply chain in an online B2C environment consists of not only the delivery system but also the upper-stream systems such as supplier network, JIT production system, distribution centers, and other logistic activities. These systems are not considered by this study and should be investigated in the future.

References

[1] Vaidyanathan BS, Matson JO, Miller DM, Matson JE. A capacitated vehicle routing problem for just-in-time delivery. IIE Transactions 1999;31(11):1083–92.

[2] Laporte G. The traveling salesman problem: an overview of exact and approximate algorithms. European Journal of Operational Research 1992a;59(2):345–58.

[3] Laporte G. The vehicle routing problem: an overview of exact and approximate algorithms. European Journal of Operational Research 1992b;59(2):231–248.

[4] Fisher ML, Jaikumar R. A generalized assignment heuristic for vehicle routing. Networks 1981;11:109–24.

[5] Bodin LD, Golden B. Classification in vehicle routing and scheduling. Networks 1981;11:97–108.

[6] Bodin L, Golden B, Assad A, Ball M. Routing and scheduling of vehicles and crews: the state of the art. Computer and Operations Research 1983;10:63–211.

[7] Wren A, Holliday A. Computer scheduling of vehicles from one or more depots to a number of delivery points. Operational Research Quarterly 1972;23:333–44.

[8] Bertsimas D, Grigni M. On the space-filling curve heuristic for the Euclidean traveling salesman problem. Operations Research Letters 1989;8:241–4.

[9] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Operations Research 1964;12:568–81.

[10] Lin S. Computer solutions on the travelling salesman problem. Bell Systems Technical Journal 1965;44:2245–69.

[11] Lin S, Kernighan B. An efficient heuristic for the traveling salesman problem. Operations Research 1973;21(2):498–516.

[12] Kolen A, Kan R, Terienekens H. Vehicle routing with time windows. Operations Research 1987;35:256–73.

[13] Powell WB, Jaillet P, Odoni A. Stochastic and dynamic networks and routing. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL, editors. Handbooks in operations research and management science, 8: network routing. Amsterdam: Elsevier Science Publishers; 1995. p. 141–295.

[14] Psaraftis HN. Dynamic vehicle routing problems. In: Golden BL, Assad AA, editors. Vehicle routing: methods and studies. Amsterdam: Elsevier Science Publishers; 1988. p. 223–48.

[15] Psaraftis HN. Dynamic vehicle routing: status and prospects. Annals of Operations Research 1995;61:143–64.

[16] Ichoua S, Gendreau M, Potvin JY. Diversion issues in real-time vehicle dispatching. Transportation Science 2000;34(4):426–38.

[17] Bertsimas DJ, Ryzin G. Stochastic and dynamic vehicle routing in the Euclidean plane with multiple capacitated vehicles. Operations Research 1993;41:60–76.

[19] Shen Y, Potvin JY, Rousseau JM, Roy S. A computer system for vehicle dispatching with learning capabilities. Annals of Operations Research 1995;61:189–211.

[20] Gendreau M, Guertin F, Potvin JY, Taillard E. Parallel Tabu search for real-time vehicle routing and dispatching. Transportation Science 1999;33:381–90.

[21] Dial B. Autonomous dial-a-ride transit: introductory overview. Transportation Research, Part C: Emerging Technologies 1995;3(5):261–75.

[22] Taillard E, Badeau P, Gendreau M, Guertin F, Potvin JY. A Tabu search heuristic for the vehicle routing problem with soft time windows. Transportation Science 1997;31(2):170–86.

[23] Potvin JY, Kervahut T, Garcia BL, Rousseau JM. The vehicle routing problem with time windows, part I: Tabu search. INFORMS Journal on Computing 1996;8(2):158–64.

[24] Potvin JY, Bengio S. The vehicle routing problem with time windows, part II: genetic search. INFORMS Journal on Computing 1996;8(2):165–72.

[25] Chiang WC, Russell RA. Simulated annealing metaheuristics for the vehicle routing problem with time windows. Annals of Operations Research 1996;63:3–27.

[26] Glover F. Tabu search part I. ORSA Journal on Computing 1989;1:190–206.

[27] Glover F. Tabu thresholding: improved search by nonmonotonic trajectories. ORSA Journal on Computing 1995;7(4):426–42.

[28] Filipec M, Skrlec D, Krajcar S. An efficient implementation of genetic algorithms for constrained vehicle routing problem. Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, 1998. p. 2231–6.

[29] Charon I, Hudy O. Application of the noising method to the traveling salesman problem. European Journal of Operational Research 2000;125:266–77.

[30] Rosenkrantz DJ, Stearns RE, Lewis PM. Approximate algorithms for the traveling salesperson problem. Proceedings of the 15th Annual Symposium on Foundations of Computer Science (FOCS 1974), 1974. p. 33–42.

[31] Rosenkrantz DJ, Stearns RE, Lewis PM. An analysis of several heuristics for the traveling salesman problem. SIAM Journal on Computing 1977;6(3):563–81.

[32] Gillett B, Miller L. A heuristic algorithm for the vehicle dispatch problem. Operations Research 1974;22:340–9.

[33] Gabow HN, Galil Z, Spencer T, Tarjan RE. Efficient algorithms for finding minimum spanning trees in undirected and directed graphs. Combinatorica 1986;6(2):109–22.

[34] Glover F. Tabu search part II. ORSA Journal on Computing 1990;2:4–32.

[35] Baykasoglu A, Gindy NN. A simulated annealing algorithm for dynamic layout problem. Computers and Operations Research 2001;28(14):1403–26.

[36] Barbarosoglu G, Ozgur D. Tabu search algorithm for the vehicle routing problem. Computers and Operations Research 1999;26(3):255–70.
[37] Baker BM, Ayechew MA. A genetic algorithm for the vehicle routing problem. Computers and Operations Research 2003;30(5):787–800.
[38] Garey MR, Johnson DS. Computers and intractability: a guide to the theory of NP-completeness. San Francisco, CA: Freeman; 1979.
[39] Croes A. A method for solving traveling salesman problems. Operations Research 1958;5:791–812.
[40] Solomon MM, Baker EK, Schaffer JR. Vehicle routing and scheduling problems with time window constraints: efficient implementations of solution improvement procedures. In: Golden BL, Assad AA, editors. Vehicle routing: methods and studies. Amsterdam: Elsevier Science Publishers; 1988. p. 85–105.
[41] Rumbaugh J. OMT insight. New York: SIGS Books; 1996.