International Journal of
Operations Research

International Journal of Operations Research Vol. 3, No. 3, 173−181 (2006)

Multi-Objective Heuristics for the Vehicle Routing Problem

Chiun-Ming Liu*, Ting-Chi Chang, and Li-Fen Huang

Department of Industrial Engineering and Systems Management, Feng Chia University, Taichung 407, Taiwan, R.O.C.

Received January 2005; *Revised November 2005*; *Accepted April 2006*

Abstract**—**The distribution service of commodities has tremendous impacts on cost-effective performance and level of customer satisfaction for third party logistics. One of main concerns is how to balance workloads among vehicles and ensure delivery time for each vehicle within the required duration. In this study, a multi-objective mixed zero-one integer programming model for the vehicle routing problem with balanced workload and delivery time are presented. In order to provide high quality solutions in short period of computational times, a heuristic-based solution method is developed. In the developed heuristic, we first generate an initial solution using savings-based procedures. Next, we devise heuristic-based procedures to improve solutions and to make sure that the workload and delivery time for each vehicle are within the limits. Then, a search heuristic procedure is used to ensure that every route is balanced in terms of workload and delivery time. An industrial size problem is applied for illustrating the proposed approach. The obtained vehicle routing schedule is better than the existing one in terms of balance in workload and delivery time among each vehicle. We also perform the computational efforts by running the developed heuristic for 12 case problems. Results suggest that the developed heuristics performs satisfactorily in terms of solution quality and execution time.

Keywords**—**Vehicle routing problem, Multi-objective optimization model, Heuristics

1. INTRODUCTION

Due to the evolution of local economy from production-oriented markets to customer-oriented markets, a large number of convenience stores and third party logistics companies established in major metropolitan areas. The distribution services of commodities provided by third party logistics to convenience stores have tremendous impacts on the cost-effective performance and the level of customer services. Thereby, the problem of how to plan and manage commodity distribution services has been received great concerns in practice.

The vehicle routing and scheduling is probably the most central model in logistics management. In the traditional vehicle routing problem, a set of convenience stores with known demand are to be served by a fleet of vehicles with known capacity. Various constraints on the routes exist. The objective is to provide services for these convenience stores in an efficient and cost-effective manner. The general vehicle routing problem could be characterized by nature of demand, information on demand, vehicle fleet, delivery time, and solution methods.

However, the attention to the local vehicle routing problem motivated by its practical relevance and considerable difficulty is how to balance workloads among vehicles and ensure delivery time for each vehicle within the required duration. We have observed numerous applications in local third party logistics where the current routes range between 3 to 8 hours in duration. The crew of fleet that works the 3 hours route may receive a full day's pay. By eliminating the imbalances in workload, delivery time, and traveling distance, overtime may be reduced, savings may be achieved, and the crews may accept the solution, because they will perceive that each crew is receiving an equal and fair deal. Hence, the special considerations that are encountered in the local thirty party logistics when solving vehicle routing problems may focus on the requirements of balanced workload and delivery time constraints in addition to the minimal total logistics cost.

In this study, the main concern is how to balance workload, delivery time, and traveling distance among vehicles in a planned vehicle routing schedule. In addition, the vehicle routing problem in this study also concerns the following aspects. We want to minimize the number of vehicles needed to provide the required distribution services and the total travel distance needed to complete the total distribution services. We also ensure that the delivery time for each routing path should be complete within a time period and the workload for every route is balanced. A multi-objective optimization model for this problem is explored and formulated. Then, a heuristic-type solution method is developed. Real-world data collected from a third party logistics is used for implementing the proposed approach. Finally, we close this paper with a description of some of our experiences in solving actual vehicle routing problems.

2. LITERATURE REVIEW

* Corresponding author's email: cmliu@fcu.edu.tw

The standard vehicle routing problem was introduced in the OR/MS literature about 45 years old. Since then, the vehicle routing problem has attracted an enormous amount of research attention.

The vehicle routing and scheduling problem was initially formulated as an integer program by Dantzig and Ramser (1959) and small problems were solved. Then in early 1960s, route-building, route-improvement, and two-phase heuristics were proposed to solve problems with 30 to 100 customers (Clarke et al., 1969). During 1970s, a number of two-phase heuristics were proposed to solve larger problems and computational efficiency became an issue (Gillett and Miller, 1974). During 1980s, mathematical programming-based procedures were proposed by Fisher and Jaikumar (1981). They showed some problems with approximately 50 customers could be solved using exact optimal methods.

In 1990s, metaheuristic-type methods, such as simulated annealing, deterministic annealing, genetic algorithm, neural networks, and tabu search, were applied for solving vehicle routing and scheduling problems (Chao et al., 1995). Some larger problems with 100 to 1,000 customers could be efficiently solved using these methods.

Recently, more research results for the algorithmic consideration are obtained. Ruiz et al. (2004) developed a decision support system for a real vehicle routing problem by using a two-stage exact approach. Firstly, they generated all the feasible solutions by means of an implicit enumeration algorithm. Then, an integer programming model was designed to select the optimum routes. Toth and Vigo (2003) devised a granular tabu search method and applied to the vehicle routing problem. Their results show that the approach is able to determine very good solutions within short computing times. Nikolakopoulou et al. (2004) developed a heuristic algorithm to balance the vehicle time utilization by partitioning a distribution network into subnetworks. Teng et al. (2003) presented three metaheuristics, simulated annealing, threshold accepting, and tabu search, for the vehicle routing problem with stochastic demand. Their results suggest that the solution quality of the tabu search outperforms the other heuristics for all the problems tested. Lysgaard et al. (2004) used a branch-and-cut algorithm for the capacitated vehicle routing problem. Their computational results show that their developed algorithm is competitive. Campbell and Savelsbergh (2004) demonstrated that insertion heuristics can be applied to solve the standard vehicle routing problem with a time complexity of $O(n(3))$. Li et al. (2005) focused on very large vehicle routing problems with many as 1200 customers by applying record-to-record travel with a variable-length neighbor list. Bazgan et al. (2005) designed constant differential approximation algorithms for the standard vehicle routing problem. Tarantillis (2005) developed an adaptive memory programming method for solving the capacitated vehicle routing problem. The developed approach provides high quality solutions in short computational times for all problems instances.

Braysy and Gendreau (2005) surveyed the past research on the metaheuristics for the vehicle routing problem with time windows. They conclude that metaheuristics are general solution procedures that explore the solution space to identify good solutions and often embed some of the standard route construction and improvement heuristics. Chepuri and Homem-de-Mello (2005) proposed a new heuristic method to solve the vehicle routing problem with stochastic demands using the cross-entropy method. Zeng et al. (2005) proposed an assignment-based local search method for solving the vehicle routing problem. Their computational results show that the proposed method, when coupled with metaheuristics such as simulated annealing, is comparable with other efficient heuristic methods. Altinel and Oncan (2005) pointed out that although classical heuristics do not compare with the best metaheuristic implementations, some of them are very fast and simple to implement. Hence, they proposed a new enhancement of the Clarke and Wright savings heuristic. Mitra (2005) developed a mixed integer linear programming formulation and a route construction heuristic for the generalized vehicle routing problem with backhauling. Funke et al. (2005) provided a review of both classical and modern local search neighborhoods for the vehicle routing problem. Their analysis shows how the properties of the partial moves and the constraints of the vehicle routing problem influences the choice of an appropriate search techniques.

In this study, the vehicle routing problem arising in local thirty party logistics is analyzed. Particularly, the following criteria are considered: (1) to minimize the number of vehicles that need to be provided the required distribution services, (2) to ensure that every route is balanced in terms of workload, and (3) to ensure that every route is balanced in terms of delivery time. Our approach is similar to the problem solved by Nikolakopoulou et al. (2004). However, more considerations are included. Moreover, a multi-objective optimization model and its associated heuristic are exploded.

3. DEVELOPMENT OF MULTI-OBJECTIVE OPTIMIZATION MODEL

The vehicle routing problem under consideration is described as follows. For a depot and many convenience stores with known locations and daily demands, we want to schedule a distribution route by a fleet of vehicles with capacity constraints and route-length constraints. Each vehicle departs from and returns to a depot to deliver a certain amount of commodities. Each convenience store is visited exactly once and its demand must be fully satisfied. The goals of this problem are to deliver the commodities to all customer at minimum total distance traveled in a set of routes without violating vehicle capacity, to minimize the number of vehicles needed to perform the required service, to ensure every route is balanced in terms of workload, and to ensure every route is balanced in terms of delivery time. In order to formulate a mathematical programming model, we first define the following variables and parameters.

Index Parameters:
I = the set for all convenience stores;
J = the set for depots;
$N = I \bigcup J$ = the set for all nodes;
K = the set for all vehicles.

Problem Parameters:
v = the maximum capacity for each vehicle;
t_{ij} = the traveling duration between node *i* and node *j*;
c_i = the unloading time for convenience store *i*;
h_i = the demand from convenience store *i*;
d_{ij} = the travel distance between node *i* and node *j*.

Decision Variables:
x_{ijk} = 1, if vehicle *k* travels from node *i* to node *j*; 0, otherwise;
y_{jk} = 1, if vehicle *k* departs from depot *j*; 0, otherwise;
wu = the maximum achieved workload among vehicles;
wl = the minimum achieved workload among vehicles;
tu = the maximum achieved delivery time among vehicles;
tl = the minimum achieved delivery time among vehicles.

A multi-objective optimization model for the vehicle routing and scheduling problem with balanced workload and delivery time may be formulated as follows.

$$\text{Minimize} \quad z = \sum_{k\in K}\sum_{i\in N}\sum_{j\in N} d_{ij}x_{ijk} \tag{1}$$

$$\text{Minimize} \quad wu - wl \tag{2}$$

$$\text{Minimize} \quad tu - tl \tag{3}$$

Subject to

$$\sum_{k\in K}\sum_{i\in N} x_{ijk} = 1 \quad \forall j \in I \tag{4}$$

$$\sum_{i\in N} x_{ijk} - \sum_{i\in N} x_{jik} = 0 \quad \forall j \in N\,; \forall k \in K \tag{5}$$

$$\sum_{i\in N} x_{ijk} = y_{jk} \quad \forall j \in J\,; \forall k \in K \tag{6}$$

$$\sum_{i\in N} x_{jik} = y_{jk} \quad \forall j \in J\,; \forall k \in K \tag{7}$$

$$\sum_{i\in I} h_i \left\{ \sum_{j\in N} x_{jik} \right\} - v \left\{ \sum_{j\in J} y_{jk} \right\} \le 0 \quad \forall k \in K \tag{8}$$

$$\sum_{i\in I} h_i \left\{ \sum_{j\in N} x_{jik} \right\} \le wu \quad \forall k \in K \tag{9}$$

$$\sum_{i\in I} h_i \left\{ \sum_{j\in N} x_{jik} \right\} \ge wl \quad \forall k \in K \tag{10}$$

$$\sum_{i\in N}\sum_{j\in N} t_{ij}x_{ijk} + \sum_{i\in I} c_i \left\{ \sum_{j\in N} x_{jik} \right\} \le tu \quad \forall k \in K \tag{11}$$

$$\sum_{i\in N}\sum_{j\in N} t_{ij}x_{ijk} + \sum_{i\in I} c_i \left\{ \sum_{j\in N} x_{jik} \right\} \ge tl \quad \forall k \in K \tag{12}$$

$$\sum_{i\in S}\sum_{j\in S}\sum_{k=1} x_{ijk} \le |S| - 1 \quad 2 \le |S| \le |I| \quad \forall S \in I \tag{13}$$

$$x_{ijk} = 0,1 \quad \forall i, j \in N\,; \forall k \in K \tag{14}$$

$$y_{jk} = 0,1 \quad \forall j \in J\,; \forall k \in K \tag{15}$$

$$wu, wl, tu, tl \ge 0 \quad \forall k \in K \tag{16}$$

The objective function (1) minimizes the total distance traveled by all vehicles. The objective function (2) is used to balance workloads among the dispatch vehicles. The objective function (3) is used to balance delivery times among the dispatch vehicles. Constraint (4) asks that every convenience store must be on exactly one route. They do so by requiring that there be exactly one node served by one vehicle preceding every convenience store. Constraint (5) is one type of flow conservation constraints. They state that if vehicle k enters node *j*, then it must depart from node *j*. Constraint (6) and (7) states that if some vehicle is assigned to a route emanating from the depot, then at least one link goes into the depot and one leaves the depot. Constraint (8) imposes the maximal workload capacity on each vehicle. Constraint (9) and (10) is used to find the maximum and minimum workloads among the vehicles. Constraint (11) and (12) is used to find the maximum and minimum delivery times among the vehicles. Constraint (13) is one type of sub-tour elimination constraints. They prevent a vehicle from being assigned to a set of nodes only. They do so by requiring that for any subset S of nodes of cardinality 2 or more, the total number of connections between pairs of nodes in the subset must be less than or equal to the cardinality of the subset minus 1. By requiring that the number of links connecting nodes in the subset be strictly less than the number of nodes in the subset, constraint (13) precludes the formation of such sub-tours. Constraint (14) and (15) is the integrality constraints. And constraint (16) is the non-negativity constraints for the maximum and minimum workloads and delivery times among vehicles.

One of the key problems in the developed optimization model is that there are a huge number of constraints. Not only are there 10 different classes of constraints, but the total number of constraints are very large even for very small problems. Constraint (13) represents most of the total number of constraints, since they apply to all subsets of the nodes. Thus, if there are $|N|$ nodes, the number of constraints in set (13) is $2^{|N|} - (|N| + 1)$. The number of constraints is not polynomially bounded as a function of the size of the problem. Hence, the presented problem is NP-hard.

Besides, the developed vehicle routing and scheduling problem is one type of multi-objective mixed zero-one programming problems. These three goals are not commensurate, which means that they cannot be directly combined or compared. It is clear that these three goals might be conflicting. That is, there are trade-offs in the sense that sacrificing the requirements on the balanced workload and delivery time goal will tend to produce greater distance on the total delivery distance. Although the existence of a solution to the vehicle routing scheduling problem depends on the number of convenience stores, the distance matrix among convenience stores and the depot and the maximal capacity of vehicles, goal programming can be used for that purpose. The way of handling multiple objectives is to reflect the desired balanced workload or delivery time a constraint, not by the objective function. Hence, we replace the objective function (2) and (3) by the following goal constraint (17) and (18).

$$wu - wl \leq w\varepsilon \tag{17}$$

$$tu - tl \leq t\varepsilon \tag{18}$$

where $w\varepsilon$ is the desired gap of tolerance between maximum and minimum workloads among vehicles and $t\varepsilon$ is the desired gap of tolerance between maximum and minimum delivery times among vehicles. The structure of the formulated goal programming problem can be expressed by the objective function (1) and constraints (4), (5), (6), (7), (8), (9), (10), (11), (12), (13), (14), (15), (16), (17) and (18).

4. DEVELOPMENT OF MULTI-OBJECTIVE HEURISTIC

The formulated optimization model is one type of mixed zero-one integer programming problems. This problem is NP-hard. Since most real-world vehicle routing problems applications are larger-scale, heuristics may be only option in practical use. In this study, a multi-objective heuristic solution approach that is capable of producing high-quality, near-optimal solutions are developed. Some of the constraints in the developed multi-objective optimization model are used as criteria to evaluate the conditions in the heuristic, such as Eq. (17) and (18). However, the procedures in the developed multi-objective heuristic are mostly based on the criteria and conditions that are demanded in the vehicle routing problem we are interested.

The developed heuristic approach consists of three phases. In the initialization phase, we generate an initial feasible solution by modifying the savings-based method. In the route improvement phase, five procedures of one-point movement, two-point exchange, intra-route arc exchange, re-initialization, and infeasibility improvement are applied to improve the obtained solution and to make sure that each vehicle's workload and delivery time is within the assigned limit. In the balancing workload and delivery time phase, we devise a searching procedure to ensure that every route is balanced in terms of workload and delivery time. The detailed procedures are given as follows.

4.1 Initialization

An initial feasible solution is constructed by modifying a savings-based method. The obtained solution satisfies the delivery time constraint and the workload capacity constraint, while reaching the minimum number of vehicles required. The detail procedure is given as follows.

Step 1. Determine the starting solution with each route for each convenience store.
Step 2. Compute the savings value for each pair of convenience stores.
Step 3. Sequence the savings values in the order of quantity.
Step 4. Select a pair of convenience stores with the largest savings value. If the selected convenience stores are not located at the end point of the route, go to *Step* 7. Otherwise, consider to join the convenience stores in one route. Proceed to *Step* 5.
Step 5. Check the workload for the jointing route. If the workload exceeds the maximum capacity, go to *Step* 7. Otherwise, proceed to *Step* 6.
Step 6. Check the time constraint for the jointing route. If the delivery time satisfies the time constraint, complete the joint of the convenience stores in one route. Proceed to *Step* 7.
Step 7. Update the savings list. If the savings list is empty, stop with a feasible solution and proceed to *Step* 8. Otherwise, return to *Step* 4.
Step 8. Compute the delivery time, the delivery distance and the workload for each route, and the total travel distance for the solution.

4.2 Route improvement

In the second phase, we adopt five procedures of one-point movement, two-point exchange, intra-route arc exchange, re-initialization, and infeasibility improvement as route improvement heuristics to obtain a satisfactory solution. These five procedures are constructed and described as follows.

4.2.1 One-point movement

The one-point movement procedure utilizes an inter-route node exchange to improve the current obtained solution. The inter-route node exchange method for the one-point movement procedure was originally proposed by Lin (1969). The idea is to remove one node from a route and then to insert into the other route using an adjacent point insertion rule. The adjacent point insertion rule determines which route and what position the inserting node should move to. We consider additional rules in this procedure to limit the capacity of vehicle and the delivery time of the route. The procedure is given as follows.

Step 0. Obtain an initial feasible solution and compute the delivery time, the delivery distance, and the workload for each route.
Step 1. Calculate the distance-savings value for each node if this node were removed from the route. Create the distance-savings list.
Step 2. Select from the distance-savings list the node with the largest distance-savings value and the associated route. Consider for removing this node from the route.
Step 3. Select the node nearest to the removing node and

not in the same route. Insert the removing node into the route that contains the node nearest to the removing node.

Step 4. Check the limits on the capacity and the delivery time for the inserting route. If these limits are not satisfied, go to *Step* 6. Otherwise, compute the increment of distance for the inserting route and proceed to *Step* 5.

Step 5. Compare the distance-savings value and the increment of distance. If the distance-savings value is greater than the increment of distance, complete the insertion and go to *Step* 7. Otherwise, proceed to the *Step* 6.

Step 6. Check the node next nearest to the removing node and not in the same route. If no such a node can be found, go to *Step* 8. Otherwise, select the route containing this node for inserting the removing node. Return to *Step* 4.

Step 7. Update the current solution and compute the delivery time, delivery distance, and workload for each route. Calculate the total distance for the current solution.

Step 8. Update the distance-savings list. If the distance-savings list is empty, stop with the obtained solution. Otherwise, return to *Step* 2.

4.2.2 Two-point exchange

The two-point exchange procedure utilizes an inter-route node exchange to improve the current obtained solution. The idea is to select two nodes in the distinct routes and then to exchange these two nodes using an adjacent point insertion rule. The adjacent point insertion rule determines which route and what position the inserting node should move to. This rule we proposed also considers the limit on the capacity of vehicle and the delivery time of the route. The procedure is given as follows.

Step 0. Obtain an initial feasible solution and compute the delivery time, the delivery distance, and the workload for each route.

Step 1. Calculate the distance-savings value for each node if this node were removed from the route. Create the distance-savings list.

Step 2. Select from the distance-savings list the node with the largest distance-savings value and the associated route. Consider for exchanging this node with another node in the distinct route.

Step 3. Pick the node nearest to the selecting node and not in the same route. Exchange the two selecting nodes.

Step 4. Check the limits on the capacity and the delivery time for the inserting route. If these limits are not satisfied, go to *Step* 6. Otherwise, compute the increment of distance for both the exchanging routes and proceed to *Step* 5.

Step 5. Compare the distance-savings value and the increment of distance. If the distance-savings value is greater than the increment of distance, complete the exchange and go to *Step* 7. Otherwise, proceed to the *Step* 6.

Step 6. Pick the node next nearest to the selecting node and not in the same route. If no such a node can be found, go to *Step* 8. Otherwise, exchange the two selecting nodes. Return to *Step* 4.

Step 7. Update the current solution and compute the delivery time, delivery distance, and workload for each route. Calculate the total distance for the current solution.

Step 8. Update the distance-savings list. If the distance-savings list is empty, stop with the obtained solution. Otherwise, return to *Step* 2.

4.2.3 Intra-route arc exchange

The intra-route arc exchange procedure utilizes a 2-opt improvement to improve the current obtained solution. The idea is to adjust the sequence of nodes in one route in order to improve the solution. During the 2-opt improvement procedure, we need to check the limit on the workload and delivery time for the route. The procedure is given as follows.

Step 0. Obtain an initial feasible solution and compute the delivery time, the delivery distance, and the workload for each route.

Step 1. Examine one route from the current solution. Select from the route a pair of arcs.

Step 2. For all of the nodes connected by this pair of arcs, reverse the sequence of delivery direction. Check the limit on the delivery time for this exchange. If the delivery time exceeds the time limit, give up this exchange and go to *Step* 4. Otherwise, proceed to *Step* 3.

Step 3. Compute the distance-savings value for this exchange. If the distance-savings value is greater than zero, complete the exchange and update the current solution. Otherwise, give up this exchange.

Step 4. If all pairs of arcs in this route are selected, go to Step 5. Otherwise, select the next pair of arcs and return to *Step* 2.

Step 5. If all routes are examined, stop with the obtained solution and compute the delivery time, the delivery distance and the workload for each route. Otherwise, examine the next route and select from this route a pair of arcs. Return to *Step* 2.

4.2.4 Re-initialization

The re-initialization procedure is used to reschedule the delivery path in the current solution in the hope to obtain a better solution. This procedure can be applied using either of the following two methods. The first method is to remove and insert a certain amount of nodes according to the distance-savings rule and the adjacent point insertion rule. The detail procedure for the first method is described as follows.

Step 1. Pick several nodes from the current solution with larger distance-savings values.
Step 2. Select the node with the largest distance-saving value from these 10 nodes as the candidate for removing and inserting.
Step 3. Pick the route with a node that is nearest to the candidate for inserting.
Step 4. Check the workload capacity and delivery time for the inserting route. If these conditions are not satisfied, give up the insertion and go to *Step* 6. Otherwise, proceed to *Step* 5.
Step 5. Compute the increment of distance. If the distance-savings value is larger than the increment of distance, complete the insertion and update the current solution. Proceed to *Step* 6.
Step 6. Select a node with the next largest distance-savings value as the candidate and return to *Step* 3. If no such a node can be found, stop with the obtained solution.

The second method is to apply the developed initialization procedure to resolve several routes in the current solution. In practice, we group the scheduled routes in the current solution into several clusters. For each cluster, the developed initialization procedure is applied to obtain a different delivery path. If the obtained solution is better that the current solution, replace the current one.

4.2.5 Infeasibility improvement

In this procedure, we apply local searching and global searching rules as well as a record-to-record travel rule. The record-to-record travel rule was proposed by Dueck (1993). The record-to-record travel rule is used to prevent from a local solution. These rules are integrated with the developed initialization procedure, the route improvement procedure and the re-initialization procedure to obtain a better solution. The local searching rule utilizes the developed intra-route arc exchange method to improve the incumbent solution. The global searching rule applies one-point movement, two-point exchange, 2-opt approach, and re-initialization to obtain a better solution. The detail solution is stated as follows.

Step 0. Apply initialization procedures to obtain a feasible solution as an incumbent solution. Compute the incumbent value.
Step 1. Apply one-point improvement procedure to search neighborhood solutions. Compute the objective value. If the obtained objective is better than the incumbent value, proceed to *Step* 2. Otherwise, go to *Step* 3.
Step 2. Apply intra-route arc exchange procedure to search neighborhood solutions. Update the incumbent solution and return to *Step* 1.
Step 3. Apply two-point exchange procedure to search neighborhood solutions. Compute the objective value. If the obtained objective is better than the incumbent value, proceed to *Step* 4. Otherwise, go to *Step* 5.
Step 4. Apply intra-route arc exchange procedure to search neighborhood solutions. Update the incumbent solution and return to *Step* 3.
Step 5. Apply one-point improvement procedure, two-point exchange procedure and intra-route arc exchange procedure to search neighborhood solutions. If the obtained solution is better than the incumbent solution, update the incumbent solution.
Step 6. Apply re-initialization procedure to search global solutions. If the obtained solution is better than the incumbent solution, proceed to *Step* 7. Otherwise, go to *Step* 8.
Step 7. Apply intra-route arc exchange procedure to search neighborhood solutions. Update the incumbent solution and return to *Step* 1.
Step 8. If the total number of looping procedures exceeds the required number, stop with a satisfactory solution. Otherwise return to *Step* 1.

4.3 Balancing workload and delivery time

In this phase, we develop a searching procedure to minimize the gap between the desired maximum workload and the desired minimum workload among the delivery routes or the vehicles. Since the customer demand should be delivered unsplitly, it is not possible to exactly balance the workload. Hence, we need to set up a tolerance of gap for balancing each vehicle's workload. The detail procedure is described as follows.

Step 0. Set the current maximal capacity for vehicles and the tolerance gap for between maximum and minimum of workload and delivery time among vehicles.
Step 1. Examine the solution obtained from the developed initialization and route improvement methods. Compute the required number of vehicles. If the required number of vehicles exceeds the previous one, go to *Step* 3. Otherwise, return to *Step* 2.
Step 2. If the calculated gap between the maximal workload and the minimal workload is less that or equal to the tolerance gap, stop with the obtained solution. Otherwise, update the current maximal capacity and proceed to Initialization phase and Route Improvement phase.
Step 3. Increase the maximal capacity for vehicles and proceed to Initialization and Route Improvement phases.

The developed heuristic procedures for the vehicle routing and scheduling problem with balanced workload and delivery time are coded by using a computer programming language, Borland C++, and compiled into an execution file. This execution file can be run on the Windows platform for solving practical problems.

5. IMPLENMENTATION

A case study using real-world data collected from a local third party logistics is illustrated. The third party logistics is located in Taichung metropolitan area, Taiwan and responsible for delivery of daily demand from a distribution center to a large number of convenience stores. The data required for the vehicle routing problem application include vehicle fleet characteristics, customer order information, and geographic data. Particularly, we collected data for travel distance among convenience stores and travel distance between the distribution center and convenience stores, daily demand from convenience stores, delivery time from the distribution center to each convenience store, and fleet information in the distribution center. These convenience stores are dispersed in the metropolitan area. The travel distance of the shortest path between two points in kilometer is measured according to the street network map and stored in a database of geography information system. The daily demand measured in volume unit is collected according to the order information.

Table 1 displays daily demand information from 81 convenience stores. The total demand is 67.47 volume units. The fleet of vehicles in this distribution center has tens of trucks, each truck with a maximum capacity of 9 units. The company asks that the workloads for each routing path should be balanced within the tolerance gap of 1.5 units. The delivery time from the distribution center to a convenience store includes the traveling time and the unloading time. The traveling time is proportionate to the distance between the distribution center and the convenience store, while the unloading time depends on the amount of ordered commodity. The company requires that the total delivery time for each routing path should be within 180 minutes. Eighty-one convenience stores are included in this problem.

In practice, the vehicle routing and scheduling problem incurred in this company has been done by experiences and rule-of-thumb. The existing vehicle schedule is shown in Table 2. In the existing vehicle schedule, the total number of vehicles needed for this distribution service is nine trucks. The total distance is 262.37 kilometers and the total distribution time is 1468 minutes. The workloads for each truck are between 8.58 and 5.46 units. The travel distances for each routing path are between 41.57 and 21.82 kilometers. The delivery times for each truck are between 193 and 127 minutes.

By applying the proposed approach to this vehicle routing problem, the obtained result is also displayed in Table 2. Eight trucks are needed for delivering the whole unsplit demand. The workloads for each truck are between 8.96 and 7.65 units. The total travel distance is 181.6 kilometers. The travel distances for each routing path are between 28.61 and 17.6 kilometers. The total delivery time is 1,219 minutes. The delivery times for each truck are between 175 and 131 minutes. The execution time for solving this problem is 2.7 CPU seconds. From Table 2 we can see that the achieved solution has a better performance in terms of total number of vehicles, total traveling distance, and total delivery time, compared with the existing routing schedule. From the viewpoint of balance in workload, traveling distance, and delivery time among each vehicle, the achieved vehicle schedule outperforms the existing one.

Furthermore, we perform the computational effort for the developed heuristic procedure in terms of computational efficiency and solution quality. We generate 12 test problems with number of nodes from 10 to 115. The test problems are generated by expanding the collected real-world data to simulate the large-scale problem. Table 3 shows the computational results. For small-size problems, the developed heuristics can provide better solutions with a much fewer CPU time. For median to large size of problems, the developed heuristics can provide good solutions within several minutes of CPU time. These results suggest that the developed solution method can be used to solve for many practical problems.

Table 1. Data of daily demand from 81 convenience stores, measured in volume unit

Store No.	Demand	Store No.	Demand	Store No.	Demand	Store No.	Demand
1	1.38	21	1.05	41	0.02	61	0.98
2	0.78	22	0.77	42	0.82	62	1.13
3	0.83	23	0.77	43	0.80	63	0.02
4	0.92	24	0.90	44	0.87	64	0.85
5	0.70	25	0.77	45	0.95	65	1.43
6	1.08	26	0.48	46	0.98	66	0.90
7	0.01	27	0.97	47	0.85	67	1.87
8	0.83	28	0.67	48	0.63	68	1.07
9	1.37	29	0.85	49	0.73	69	0.73
10	1.20	30	0.98	50	0.90	70	0.65
11	1.17	31	0.75	51	0.75	71	0.62
12	0.87	32	0.77	52	0.01	72	0.60
13	0.53	33	0.80	53	0.90	73	0.68
14	1.20	34	0.60	54	0.90	74	0.72
15	0.62	35	1.28	55	0.90	75	1.12
16	0.63	36	0.92	56	1.08	76	1.03
17	0.63	37	0.63	57	0.65	77	0.65
18	1.25	38	0.55	58	0.77	78	0.90
19	0.58	39	0.57	59	1.35	79	1.03
20	0.82	40	0.57	60	0.90	80	0.88
						81	0.80

Table 2. Comparison of results for the existing routing schedule and the achieved routing schedule

Alternative	Vehicle No.	Routing Path for Each Vehicle	Delivery Time (min)	Delivery Distance (km)	Workload (unit)
The Existing Vehicle Routing Schedule	1	0⇨1⇨6⇨28⇨7⇨11⇨24⇨25⇨30⇨23⇨ 29⇨0	183	30.87	8.58
	2	0⇨14⇨5⇨19⇨17⇨12⇨18⇨31⇨4⇨3⇨2⇨0	150	30.80	8.51
	3	0⇨13⇨10⇨9⇨26⇨8⇨22⇨21⇨15⇨20⇨38⇨0	152	41.57	8.22
	4	0⇨36⇨37⇨39⇨53⇨71⇨63⇨16⇨61⇨62⇨60⇨0	181	27.52	7.30
	5	0⇨35⇨46⇨47⇨48⇨45⇨32⇨0	127	23.34	5.46
	6	0⇨40⇨69⇨49⇨51⇨50⇨54⇨70⇨55⇨72⇨0	150	26.69	6.73
	7	0⇨44⇨41⇨27⇨43⇨42⇨56⇨58⇨59⇨57⇨0	166	21.82	7.33
	8	0⇨64⇨65⇨76⇨68⇨67⇨66⇨52⇨34⇨33⇨0	166	23.88	8.56
	9	0⇨73⇨77⇨81⇨75⇨79⇨78⇨80⇨74⇨0	193	35.88	6.78
	Total		1468	262.37	67.47
The Vehicle Routing Schedule Achieved by the Proposed Heuristic	1	0⇨1⇨10⇨9⇨11⇨4⇨3⇨2⇨0	132	18.48	7.65
	2	0⇨12⇨31⇨32⇨46⇨61⇨62⇨34⇨47⇨35⇨16⇨0	154	21.95	8.84
	3	0⇨29⇨44⇨57⇨59⇨58⇨56⇨71⇨70⇨69⇨38⇨0	157	25.17	8.12
	4	0⇨5⇨14⇨18⇨19⇨17⇨15⇨21⇨6⇨8⇨0	132	17.60	7.94
	5	0⇨7⇨22⇨27⇨28⇨42⇨43⇨30⇨26⇨25⇨24⇨23⇨13⇨0	148	21.18	8.47
	6	0⇨36⇨48⇨65⇨80⇨81⇨73⇨72⇨54⇨55⇨41⇨40⇨37⇨0	175	28.61	8.96
	7	0⇨39⇨49⇨50⇨51⇨53⇨68⇨67⇨66⇨52⇨20⇨0	147	20.74	8.52
	8	0⇨64⇨63⇨76⇨77⇨78⇨79⇨75⇨74⇨60⇨45⇨33⇨0	173	27.87	8.97
	Total		1218	181.60	67.47

Table 3. Computational results by running the proposed heuristic

Prob. No.	No. of Nodes	Max. Capacity	Limit on Delivery Time (min.)	No. of Trucks	Heuristic	
					Total Distance (km)	CPU Time (sec)
1	10	10.5	120	2	28.38	1.64
2	20	10.5	120	3	53.98	4.98
3	30	10.5	120	5	87.80	3.94
4	40	10.5	120	6	111.59	5.70
5	50	10.5.	120	7	135.73	2.65
6	60	10.5	120	9	179.16	2.85
7	70	10.5	120	11	223.71	16.91
8	80	15.5	120	13	273.74	12.43
9	90	15.5	120	13	272.15	16.14
10	100	15.5	120	15	229.19	77.92
11	110	15.5	120	16	230.85	37.73
12	115	15.5	120	17	268.50	186.11

6. CONCLUSIONS

The distribution services of commodities provided by third party logistics have tremendous impacts on the cost-effective performance and the level of customer satisfaction. The planning of commodity distribution have been received great concerns in practice. In this paper, we describe the goals for a vehicle routing problem and show how these goals may differ from the goals of traditional routing and scheduling problems. A multi-objective optimization model for describing the vehicle routing and scheduling problem with balanced workload and delivery time is presented. The model can be used to minimize the number of vehicles, to minimize the total travel distance needed to complete the distribution service, to ensure that every route is balanced in terms of workload and delivery time. A heuristic-type solution method is developed. Real-world data collected from a local third party logistics are used for illustrating the proposed approach. The achieved routing schedule provides smaller total travel distance, shorter total delivery time, smaller number of trucks required, and balanced workload and delivery time. Results from computational effort also suggest that the developed solution method can be used to efficiently solve median to large-scale practical vehicle routing problems.

The significance and originality of our study are at least two-fold. One is to construct a multi-objective optimization model for representing the vehicle routing problem with balanced workload and delivery time. The formulated model possesses its own originality in nature. The other is to propose a heuristic method for solving the problem. Especially, in the proposed heuristic, we modified and integrated five procedures, one-point movement, two-point exchange, intra-route arc exchange, re-initialization, and infeasibility improvement, to improve the obtained solutions. The computational results show its significance in terms of efficiency.

ACKNOWLEDGEMENTS

This study was supported by the National Science Council of Executive Yuan, Taiwan. (NSC 89 – 2213 – E – 035 – 009)

REFERENCES

1. Altinel, I.K. and Oncan, T. (2005). A new enhancement of the Clarke and Wright savings heuristic for the capacitated vehicle routing problem. *Journal of the Operational Research Society*, 56(8): 954-961.
2. Bazgan, C., Hassin, R., and Monnot, M. (2005). Approximation algorithms for some vehicle routing problems. *Discrete Applied Mathematics*, 146(1): 27-42.
3. Braysy, I. and Gendreau, M. (2005). Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation Science*, 39(1): 119-139.
4. Campbell, A.M. and Savelsbergh, M. (2005). Efficient insertion heuristics for vehicle routing and scheduling problems. *Transportation Science*, 38(3): 369-378.
5. Christofides, N. and Eilon, S. (1969). An algorithm for the vehicle distribution problem. *Operational Research Quarterly*, 20: 309-318.
6. Clarke, G. and Wright, J.W. (1964). Scheduling of vehicles from a center depot to a number of delivery points. *Operations Research*, 11: 568-581.
7. Chao, I.M., Golden, B., and Wasil, E. (1995). An improved heuristic for the periodic vehicle routing problem. *Networks*, 26: 25-44.
8. Chepuri, K. and Homem-de-Mello, T. (2005). Solving the vehicle routing problem with stochastic demands using the cross-entropy method. *Annals of Operations Research*, 134(1): 153-181.
9. Dantzig, G. and Ramser, J. (1959). The truck dispatching problem. *Management Science*, 6: 81-91.
10. Dueck, G. (1993). New optimization heuristics: the great deluge algorithm and the record-to-record travel. *Journal of Computational Physics*, 104: 86-92.
11. Fisher, M.L. and Jaikumar, R. (1981). A generalized assignment heuristics for vehicle routing. *Networks*, 11: 109-124.
12. Funke, B., Grunert, T., and Irnich, S. (2005). Local search for vehicle routing and scheduling problems: Review and conceptual integration. *Journal of Heuristics*, 11(4): 267-306.
13. Gillett, B.E. and Miller, L.R. (1974). A heuristic algorithm for the vehicle dispatch problem. *Operations Research*, 22: 340-349.
14. Li, F.Y., Golden, B., and Wasil, E. (2005). Very large-scale vehicle routing: new test problems, algorithms, and results. *Computers & Operations Research*, 32(5): 1165-1179.
15. Lin, S. (1965). Computer solutions of the traveling salesman problem. *The Bell System Technical Journal*, 2245-2269.
16. Lysgaard, J., Letchford, A.N., and Eglese, R.W. (2004). A new branch-and-cut algorithm for the capacitated vehicle routing problem. *Mathematical Programming*, 100(2): 423-445.
17. Mitra, S. (2005). An algorithm for the generalized vehicle routing problem with backhauling. *Asia-Pacific Journal of Operational Research*, 22(2): 153-169.
18. Nikolakopoulou, G., Kortesis, S., Synefaki, A., and Kalfakakou, R. (2004). Solving a vehicle routing problem by balancing the vehicles time utilization. *European Journal of Operational Research*, 152(2): 520-527.
19. Ruiz, R., Moroto, C., and Alcaraz, J. (2004). A decision support system for a real vehicle routing problem. *European Journal of Operational Research*, 153(3): 593-606.
20. Tarantilis, C.D. (2005). Solving the vehicle routing problem with adaptive memory programming methodology. *Computers & Operations Research*, 32(9): 2309-2327.
21. Teng, S.Y., Ong, H.L., and Huang, H.C. (2003). A comparative study of metaheuristics for vehicle routing problem with stochastic demands. *Asia-Pacific Journal of Operational Research*, 20(1): 103-119.
22. Toth, P. and Vigo, D. (2003). The granular tabu search and its application to the vehicle-routing problem. *INFORMS Journal on Computing*, 15(4): 333-346.
23. Zeng, L., Ong, H.L., and Ng, K.M. (2005). An assignment-based local search method for solving vehicle routing problems. *Asia-Pacific Journal of Operational Research*, 22(1): 85-104.