
Tsai-Yun LIAO  
Associate Professor  
Graduate Institute of Marketing and Logistics/Transportation  
National Chiayi University  
No.580, Sinmin Rd., Chiayi City 60054  
Taiwan (R.O.C.)  
Fax: 886-5-2732932  
Email: tyliao@mail.ncyu.edu.tw

Shiang-Jeng WANG  
Graduate Student  
Transportation and Communication Management Science  
National Cheng Kung University  
No.1, University Road, Tainan City 701  
Taiwan (R.O.C.)  
E-mail: wxz159d2@hotmail.com

Ta-Yin HU  
Professor  
Transportation and Communication Management Science  
National Cheng Kung University  
No.1, University Road, Tainan City 701  
Taiwan (R.O.C.)  
Fax: +886-6-2753882  
E-mail: tyhu@mail.ncku.edu.tw

Abstract: The on-line vehicle routing problems (VRP) is an extension of VRP in order to consider real-time requests as well as possible variations travel times in the network. In this research, a solution algorithm for solving on-line VRP is proposed. The solution algorithm is divided into two phases, off-line route planning and on-line route updating. In the off-line phase, a time-dependent VRP formulation is constructed to assign initial routes. In the on-line routing phase, a hybrid heuristic approach with tabu search and genetic algorithm is proposed to handle real-time requests and to improve routes under real-time information. The simulation-assignment model, DynaTAIWAN is applied to evaluate assigning and routing strategies in a traffic network. Numerical experiments are conducted in a Kaohsiung city network.

Key Words: on-line VRP, hybrid heuristic approach, tabu search, genetic algorithm

1. INTRODUCTION

City logistics management has been one of the most important parts in the supply chain management. It consists of the process of totally optimizing urban logistics activities by considering the social, environmental, economic, financial, and energy impacts of urban freight movement (Taniguchi et al., 2001). More and more dispatching centers (DC) have been established to provide the flexibility of pickups and deliveries. Major daily operation issues in distribution centers are routes and schedules of trucks. Due to the advancement of information and communication, city logistics operations could be designed more efficiently, such as on-line operations. The technology applications in Commercial Vehicle Operations (CVO), especially in communication and information technologies, allow the study of on-line vehicle routing problems (VRP) under new and updated information, such as real-time traffic conditions, vehicle status, and new coming demands (Psaraftis, 1995; Gendreau et al., 1999; Hu et al., 2003; Ghiani et al., 2003). In order to dispatch commercial vehicles more
efficiently in city logistics management, the solution algorithm for the on-line VRP problems is designed to re-schedule planned routes to account for the occurrence of new customer requests and dynamic travel times.

Two major operational benefits of CVO include: (1) dynamically assign vehicles to time-sensitive demands, and (2) efficiently reroute vehicle according to current traffic conditions. Critical problems in vehicle dispatching include vehicle assignment and vehicle routing problems in real time. Vehicle assignment and routing problems have been studied for several decades (Bodin et al., 1983; Powell and Spivey, 2004). Although most real-world VRP are dynamic, and the traditional methodologies for this class of problems has been based on adaptations of static algorithms. These routing strategies are developed under known demands and static travel time, but they do not consider real-time requests and traffic flow conditions. On-line VRP need to consider real-time information as well as demands, and thus information attributes are important.

The dynamic VRP considers some variations of input data, such as demand, load, and travel time in different ways. However, the on-line VRP is an extension of VRP in order to consider possible variations of real-time demands and travel times in the network. A vehicle fleet of fixed capacities has to serve customers with time-window constraints of fixed demand from a central depot. Customers must be assigned to vehicles and the vehicles are routed to serve these demands within time-window constraints so that the total time spent on the routes is minimized.

In this research, a solution algorithm for solving on-line VRP is proposed. The solution algorithm is divided into two phases, off-line route planning and on-line route updating. In the off-line phase, a time-dependent VRP formulation is constructed to assign initial routes. The time-dependent travel times are modeled based on the concept of step functions, thus variations of link travel time could be captured. In the on-line routing phase, a hybrid heuristic approach is proposed to handle real-time requests and to improve routes under real-time information. The hybrid approach is developed based on tabu search (TS) and genetic algorithm (GA).

The proposed solution algorithms are tested through numerical experiments in an evaluation framework in which assigning and routing operations could be simulated in a realistic traffic environment. The simulation-assignment model DynaTAIWAN (Hu et al., 2007) is applied to evaluate assigning and routing strategies in a traffic network. Numerical experiments are conducted in a Kaohsiung city network to explore the proposed algorithm for dynamic fleet management problem under real-time information supply strategies.

The major contributions of the research include (1) the consideration of real-time request and real-time travel cost simultaneously; (2) the development of the mixed meta-heuristic algorithm to improve efficiency.

This paper is organized as follows: the next section briefly describes some related research. The research framework and solution algorithms are discussed in the third section. Numerical experiments and analysis are discussed in the fourth section. Concluding comments are given in the last section.
2. LITERATURE REVIEW

The fleet management problem includes two types of sub-problems: fleet assignment problem and routing problem (Bramel and Simchi-Levi, 1997). Assume a dispatcher uses a fleet of vehicles of limited capacity to serve a set of demands. First, the dispatcher must decide how to partition the demands into groups that can be served by a vehicle. Second, the dispatcher must decide what sequence to use so as to minimize cost. VRP is the problem of constructing vehicle routes of minimum total cost starting and ending at a depot, such that each node is visited by one vehicle, and satisfying some constraints, such as capacity, duration, and time windows. Since the VRP problem is NP-hard, different solution techniques, including heuristics, mathematical programming based heuristics, meta-heuristics, and polyhedral combinatorics based optimization algorithms, are applied to obtain acceptable solutions within a reasonable time frame.

For dynamic assignment problem, Powell and Spivey (2004) describe a set of network assignment models, from deterministic to stochastic models, and also presents a hybrid model. Brown and Graves (1981) present an integer linear programming formulation of a real-time routing and scheduling problem for petroleum tank trucks. The model develops truck tours for known (deterministic) customer demands. Gavish (1981) describes an optimization-based, hierarchical model for real-time routing scheduling. All of these papers present optimal algorithms or near optimal heuristics under real-time environment. The underlying models do not incorporate forecasted demands and/or real travel time.

With the advancement of communication and information technologies, real-time traffic conditions as well as dynamic demands are possible to obtain during the vehicle’s journey, thus a realistic VRP is defined as dynamic vehicle routing problems (DVRP). Powell(1995) classified the DVRP as deterministic and dynamic problems. In the deterministic problems, all data are known in advance; however, in the dynamic problems, some elements of information are revealed with respect to time. In stochastic and dynamic problems VRP, uncertain data are represented by stochastic processes that will be a real-time VRP (Ghiani et al., 2003). Psaraftis (1995) has addressed some basic characteristics of DVRP, and pointed out that computer and communication technologies, such as electronic data interchange (EDI), geographic information systems (GIS), global positioning systems (GPS), and Intelligent Transportation Systems (ITS), have significantly enhanced the possibilities for efficient dynamic routing. Possible information attributes might include evolution of information (static/dynamic), quality of information (known-deterministic /forecast /probabilistic /unknown), availability of information (local/global), and processing of information (centralized/decentralized) (Psaraftis, 1995). These information attributes might have great impact on how to develop and design an efficient and good dynamic vehicle routing algorithm. Under on-line VRP consideration, Ghiani et al. (2003) list several possible applications for this type of problems: dynamic fleet management, couriers, rescue and repair service companies, dial-a-ride system, taxi cab service, and emergency services. In the on-line VRP, Lorinia et al. (2011) suggested that the online planned routes can be quickly modified to account for the occurrence of new customer requests, and they proposed a modified approach for VRP with dynamic requests and dynamic travel times.

In order to consider travel time variations, different approaches have been developed. Malandraki and Daskin (1992) used a step function to represent time-dependent issue and develop a heuristic approach. Stochastic VRP (SVRP) have been proposed to consider such travel time variations (Gendreau et al., 1996; Laporte, et al., 1992). Due to the difficulties of
capturing the variation of travel time in a traffic network, simulation models have been used to generate realistic travel time and applied in different routing strategies. Hu (2001) provides an evaluation framework under the consideration of real-time information, and the vehicle routing strategies are solved through a heuristic approach. Hu et al. (2003) applied a SVRP approach in a chance-constrained formulation (Laporte et al., 1992) and the SVRP solution formulation is solved through branch-and-bound technique by CPLEX. Ichoua et al. (2003) propose time-dependent vehicle speed model to consider possible travel time variation for different time intervals and links, and the approach is solved through a parallel TA heuristic. Fleischmann et al. (2004) use dynamic information and dynamic path calculation, and propose three different heuristics to solve the problem.

According Laporte et al. (2000), TA heuristics have proved to be the most successful metaheuristic approach. The TA algorithm, a memory-based search strategy, attempts to guide the local search method to continue its search beyond a local optimum. The algorithm keeps a tabu list of moves or solutions that have been made or visited in the past. The purpose of the tabu list is to record a number of most recent moves and to prohibit any repetition or cycling. A number of researchers have applied the TA algorithm on VRP. Gendreau et al. (1999) propose a TA heuristic approach to DVRP and implement on a parallel computer platform to increase the computational effort. Liao (2004) addresses the DVRP problem and implements a TA heuristic algorithm.

A hybrid heuristic approach combines several meta-heuristic approaches to enhance solution accuracy and efficiency. Several hybrid algorithms have been proposed to improve efficiency and accuracy, and to avoid local optimal solutions. Ribeiro et al. (2005) proposed Iterated Local Search (ILS) and Variable Neighborhood Search (VNS) to solve car sequencing problems. Mauri and Lorena (2006) used Population Training Algorithm (PTA) and linear programming to solve crew scheduling problems. Ho et al. (2008) solved the multi-depot vehicle routing problem (MDVRP) by two hybrid genetic algorithms (HGA): HGA1 and HGA2. HGA1 is a random base initial method GA and HGA2 is a nearest neighbor heuristic initial GA. The results of HGA2 are superior to those of HGA1 in terms of the solutions’ quality. Lin et al. (2009) proposed a hybrid algorithm which takes the advantages of simulated annealing (SA) and TS to compare the results of benchmarks for capacitated vehicle routing problems (CVRP). Wang and Lu (2009) solved the CVRP with HGA and SA.

GA and TS are two well-known heuristic algorithms. Studies have been shown that both the algorithms can achieve good results in combinatorial optimization problems. Various studies have tried to combine TS and GA to solve complex problems. Glover et al. (1995) proposed the scatter search to provide possibilities for integrating GA and TS. Thangiah et al. (1994) combined GA, SA and TA to solve the vehicle routing problem with time windows (VRPTW). Their numerical experiments are selected from Solomon’s benchmarks (Solomon, 1987), and the computational results obtain new best solutions for 40 test problems and optimal number of vehicles. Ting et al. (2001) proposed a tabu search with genetic algorithm (TGA) which integrates the memory structure and search strategy of TS with GA. Their results show that the selection efficiency is improved and the population diversity is maintained by incorporating the regeneration operator.

3. RESEARCH FRAMWORK AND SOLUTION ALGORITHM

In this research, a solution algorithm for solving on-line VRP is proposed. The whole process
is illustrated in Figure 1. The solution algorithm is divided into two phases, off-line route planning and on-line route updating. In the off-line phase, the routing strategies are designed off-line and a time-dependent VRP formulation is constructed to assign CVO vehicles with initial routes. The time-dependent travel times are modeled based on the concept of step functions, thus variations of link travel time could be captured. In the on-line routing phase, a hybrid heuristic approach, developed based on tabu search and genetic algorithm (Tabu-GA), is proposed and implemented to serve real-time demands and improve routes under real-time information.

Figure 1 The flow chart of solution algorithm for on-line VRP

3.1 Time-dependent VRP Formulation
The time-dependent travel times are modeled based on the concept of step functions, thus variations of link travel time could be captured. The time-dependent VRP can be formulated as follows and the objective is to minimize the total CVO travel time.
Min \( \sum_{k=1}^{K} t_{n+k} - \sum_{i=2}^{n} t_{i} \)  \( (1) \)

subject to

\[ \sum_{i=2}^{n} \sum_{m=1}^{M} x_{ij}^{m} = 1, \quad j = 2...n + K, \quad i \neq j \] \( (2) \)

\[ \sum_{j=2}^{n} \sum_{m=1}^{M} x_{ij}^{m} = 1, \quad i = 2...n, \quad i \neq j \] \( (3) \)

\[ \sum_{i=2}^{n} \sum_{m=1}^{M} x_{ij}^{m} = K \] \( (4) \)

\[ \sum_{m=1}^{M} x_{i1}^{m} = 1, \quad k = n+1...n+K \] \( (5) \)

\[ t_{j} - t_{i} - Bx_{ij}^{m} \geq c_{ij}^{m} - B \] \( (6) \)

\[ t_{j} - t_{i} + Bx_{ij}^{m} \leq c_{ij}^{m} + B \] \( (7) \)

\[ t_{j} - T_{m}x_{ij}^{m} \geq 0 \] \( (8) \)

\[ t_{j} + Bx_{ij}^{m} < T_{m+1} + B \quad i = 1...n; \quad j = 2...n; \quad i \neq j; \quad m = 1...M \] \( (9) \)

\[ t_{i} \leq U_{i}, \quad i = 2...n \] \( (10) \)

\[ w_{j} - w_{i} - B \sum_{m=1}^{M} x_{ij}^{m} \geq d_{j} - B, \quad i = 1...n; \quad j = 2...n; \quad i \neq j, \] \( (11) \)

\[ w_{n+k} \leq b_{k}, \quad k = 1...K \] \( (12) \)

\[ w_{1} = 0 \] \( (13) \)

\[ x_{ij}^{m} = \{0,1\}; \quad \forall i, j, m \] \( (14) \)

\[ w_{i} \geq 0; \quad t_{i} \geq 0 \quad , \quad i = 1...n \] \( (15) \)

The indexes, parameters, and decision variables used in the model are described as follows.

1. Indexes:
   - \( i, j \): nodes, \( i=1 \) is depot, \( i,j=2...n \) imply there are \( n-1 \) demands.
   - \( k \): the set of CVO vehicles, \( k=1...K \).
   - \( m \): time intervals, \( m=1...M \).

2. Parameters:
   - \( c_{ij}^{m} \): travel time from node \( i \) to node \( j \) during time interval \( m \).
   - \( T_{m} \): the start time for time interval \( m \).
   - \( d_{i} \): demand of node \( i \).
   - \( b_{k} \): the capacity of CVO vehicle \( k \).
   - \( U_{i} \): the latest time that node \( i \) can be serviced.
   - \( B \): a big number.

3. Decision variables:
   - \( x_{ij}^{m} \): 1 if any CVO travels from node \( i \) to node \( j \) during time interval \( m \), 0 otherwise.
   - \( t_{i} \): departure time of any CVO vehicle from node \( i \).
   - \( w_{i} \): amount of load carried by a CVO when departing from node \( i \).
The objective function tries to minimize the travel route time of all CVO. The first term of the objective function, \( \sum_{k=1}^{x} t_{n+k} \), proposed by Malandraki and Daskin (1992), is used and designed with virtual nodes from \( n+1 \) to \( n+K \) to represent the travel time of CVO to virtual nodes. In order to reflect the time-window, the lower bound constraint is relaxed and added in the objective function, thus the second term \( \sum_{i=2}^{n} t_i \) is the lower-bound penalty for soft-time window considerations. The design of second term pushes the CVO to serve demand nodes as late as possible, but still stratifies the upper bound constraints. The upper bound constraint is still kept in the formulation. Constraints (2) and (3) are flow conservation constraints and ensure each node serviced by only one CVO. Constraints (4) and (5) ensure that the CVO vehicles return to the depot. Constraints (6) and (7) are travel time constraints for the CVO vehicles; when the CVO travels from \( i \) to \( j \) at time interval \( m \), the travel time equal to the route’s travel time at interval \( m \). Constraints (8) and (9) calculate \( t_i \). When the CVO travels from \( i \) to \( j \) at time interval \( m \), the departure time of node \( i \) is between \( T^m \) to \( T^{m+1} \). Constraint (10) is the time window constraint. Constraint (11) implies that the increase of vehicle load from \( i \) to \( j \) is equal to the demand of node \( i \). Also, it is served as a sub-tour constraint. Constraint (12) is the capacity constraint. Constraint (13) initiates the empty load of each CVO at depot. Constraints (14) and (15) specify the integrality of the route variables and non-negativity for vehicle loads and departure time variables.

The time-dependent VRP formulation is solved through CPLEX, a mathematical programming software. The performance of the formulation is discussed and compared in the numerical experiments.

### 3.2 The Hybrid Heuristic Approach

In this research, a hybrid algorithm Tabu-GA is employed in dealing with on-line requests and real-time rerouting to efficiently optimize route strategies. The proposed algorithm uses TS algorithm to define search directions and deploys GA algorithm to refine local search. The GA generates better solutions by crossover and mutation at each iteration (or generation) of an entire solution process. Generating offspring of parents, which have been generated up to the previous iteration, corresponds to choosing better solutions among the candidate solutions.

The chromosome (gene) of the candidate solution is designed and illustrated in Figure 2. The chromosome represents the coding of a combination of nodes and associated route sequence. Each chromosome is assigned a unique ID, fitness value, and tabu lists. The candidate gene is illustrated in Figure 3.

<table>
<thead>
<tr>
<th>Gene #</th>
<th>fitness( )</th>
<th>Tabu list</th>
<th>Gene codes of routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>100</td>
<td>3 12 9</td>
<td>0  D2  D3  D5  D7  0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>1 3 7</td>
<td>0  D2  D23  D5  D7  0</td>
</tr>
</tbody>
</table>

Figure 2 Gene code structure

<table>
<thead>
<tr>
<th>fitness( )</th>
<th>Parents' Gene #</th>
<th>Offspring gene codes of routes</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>9 22</td>
<td>0  D2  D3  D5  D7  0</td>
</tr>
<tr>
<td>80</td>
<td>13 8</td>
<td>0  D2  D23  D5  D7  0</td>
</tr>
</tbody>
</table>

Figure 3 Candidate gene code structure
A fitness value is used for choosing two parents to apply the crossover operator. The fitness value reflects the goodness of an individual, compared with the other individuals in the populations. The selection efficiency is improved and the population diversity is maintained by incorporating the regeneration operator. In this study, the fitness function is the inverse of the total travel cost. The function is described as follows:

$$fitness(S) = \frac{1}{TCost(S)}$$

(16)

By favoring better solutions, the selection procedure designed in the Tabu-GA algorithm guides the search toward high performance regions of the search space. Individuals in the mating pool are paired to produce offspring. The crossover is recorded in the tabu list to prohibit repetitive reproduction in the crossover process.

The effect of the tabu list in the algorithm focuses on the diversification in population without sacrificing the intensification. As the tabu list prevents some moves from trapping in local optimum in TS, it forbids the individuals with same clan to mate in the Tabu-GA. Similar to the move in the TS, the selection is restricted by tabu list; i.e., mating with a chromosome labeled tabu is forbidden until the chromosome is removed from the tabu list. The aspiration criterion provides opportunity for the superior solution to override the tabu restriction, otherwise the chromosomes are added in the tabu list. The candidate solutions are generated through the crossover process. Each candidate solution is constructed with the solution, the parent and fitness values.

The algorithm is illustrated in Figure 4 and algorithmic steps are summarized as follows:

Step 1. Initialization:
Each gene in the initial population is assigned to random numbers by independently setting each bit value to either 0 or 1 with equal probability. Then, a unique ID is assigned and the associate fitness value is calculated.

Step 2. Selection and crossover:
Individuals in the population are selected for reproduction. Let \( N_{pop} = \) number of individuals in the population and \( f_i = fitness \) value for the \( i \)th individual. The probability that the \( i \)th individual is selected for production is proportional to the fitness value. Then the chromosome are generated and added in the candidate list through crossover and mutation. The process is repeated till the candidate list is full.

Step 3. Candidate list:
Select the best solution from the candidate list and examine the tabu list to check whether the solution is in the tabu list. If the solution is in the tabu list and the solution is superior, apply the aspiration criterion to remove the solution from the tabu list; otherwise go back to Step 2.

Step 4. Individuals in the mating pool are paired to produce offspring. The new offspring is assigned with a unique ID. The gene also records their parent gene IDs then added in the population and return to step 2.

Step 5. Replacement:
The individuals in the initial population constitute the first generation. After selection, crossover, and mutation, a new population or generation of individuals is formed. In each generation, the inferior chromosomes are removed based on elimination cycle (EC) and elimination percentage (EP%).

Step 6. Convergence criteria:
When the number of generation reaches the pre-defined number, the algorithm is stopped. Otherwise, go to step 2.
The hybrid Tabu-GA algorithm is implemented in a simulation framework. The simulation framework is implemented in the object-oriented approach, and four major modules are developed. DynaTAIWAN, an integrated dynamic simulation-assignment model, is applied to construct the simulation platform to generate realistic traffic environments and evaluate the proposed routing algorithm through vehicle simulation in the traffic network. The overall program structure is shown in Figure 5. These modules are described hereafter:

1. Input module
In the simulation framework, required data sets include simulation data and VRP data. The simulation data includes network data, traffic control data, and time-dependent O-D trip tables. The VRP data include number of off-line and on-line demand nodes to be served and the location of the depot. The time-dependent O-D trip tables and network data are primarily used in the simulation to reflect realistic traffic environment.
2. Traffic simulation module
The main purpose of the module is to simulate vehicle’s movements in realistic traffic conditions. Simulated vehicles include passenger cars, motorcycles, and commercial vehicles. Passenger cars and motorcycles are generated through O-D tables and commercial vehicles are generated according to the VRP data. During the simulation, link travel costs are estimated for every simulation interval and a travel time matrix from node to node is calculated based on the Floyd-Warshall algorithm, the all pair shortest path algorithm. The travel time matrix is used in the route updating module to calculate new routes for CVO vehicles.

3. Route updating module
The off-line and on-line routing strategies are generated in this module. The time-dependent VRP formulation is solved off-line through CPLEX outside of the framework. The on-line phase uses the hybrid algorithm to review on-line requests and to re-calculate optimal routes for CVO vehicles based on real-time travel costs. The algorithm updates service sequence and paths between each pair of demand nodes.

4. Output module
The module keeps track updated routes and travel time information for each vehicle.
4. NUMERICAL EXPERIMENTS AND RESULTS

4.1 Experimental Setups
In the numerical experiments, the study area covers the sub-network of the Kaohsiung City, the largest metropolitan in southern Taiwan. The population in the network is about 350,000, and about 1/4 of the total population in the Kaohsiung city. As shown in Figure 6, the network includes 363 links and 132 nodes, and 27 traffic zones, including 5 external zones. The O-D demand in the experiment is based on the project, titled of “Comprehensive Planning of Light Rail Project in Kaohsiung Metropolitan Area” (Sinotech Engineering Consultants, 2007).

![Figure 6 The test network: a sub-network of Kaohsiung City](image)

In the parameter setting, different scenarios are designed to observe the performance of the proposed algorithm. The depot and 20 demands are randomly chosen from the network. Each demand is assigned a time-window with the earliest service time and the latest service time. The earliest service time is randomly generated with 20 minutes to 140 minutes and the latest service time is generated with the range of 20 to 80 minutes after the earliest service time. Among these 20 demands, 10 demands are randomly chosen for real-time release and their release times are randomly generated with the simulation time and the earliest service time.

Several performance indexes are chosen to measure the proposed algorithm, and these performance indexes are summarized in Table 1. There are three important aspects, cost, level of service, and computational efficiency.
Table 1 Performance indexes

<table>
<thead>
<tr>
<th>Classification</th>
<th>Performance index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distribution cost</td>
<td>Total CVO travel time</td>
</tr>
<tr>
<td></td>
<td>Number of CVO vehicles</td>
</tr>
<tr>
<td>Level of Service</td>
<td>Violation of Vehicle Capacity</td>
</tr>
<tr>
<td></td>
<td>Violation of time-window constraint</td>
</tr>
<tr>
<td>Computational efficiency</td>
<td>CPU time</td>
</tr>
</tbody>
</table>

4.2 Basic Experiments and Parameter Setting

Several important parameters in the proposed algorithm are examined through numerical experiments. The parameters and designed values are described as follows:

1. Real-time updating frequency: 5 minutes (300 seconds). The real-time requests and routing strategies are updating every 5 minutes.

2. Stopping criteria:
   (1) The maximum number of generations: 300. The value basically determines the computation iterations and time.
   (2) Maturation diversity range: 10%. The alternative stop rule is that when the population is mature, which is defined as the range of fitness values are within a diversity range of 10%.

3. TS:
   (1) Candidate list: 100. The candidate list determines the number of generated chromosomes for each generation. The search accuracy highly depends on the list.
   (2) Tabu list: 10. The tabu list records recent generated chromosomes and prohibits search toward these used chromosomes. In this case, the TA could increase the diversity in later generations.

4. GA:
   (1) Initial population: 100. If the number of initial solution is larger, more chromosomes are considered in the initial step.
   (2) Mutation rate: 0.01. Mutation is performed by flipping each bit in the offspring independently with probability $P_m$. Mutation induces random alternations to the genetic materials. If the alternation introduced is desirable, then the individual will have a higher chance of passing on its genetic materials.
   (3) Elimination cycle: 10.
   (4) Elimination percentage: 10%.

5. Penalty values:
   (1) Penalty for abandoning the new request: 10 minutes (600 seconds). When the dispatching center is unable to handle a new request, the request is abandoned and the penalty is added in the objective value.
   (2) Penalty for time-window violation: 5 minutes (300 seconds). If the value is high, the tendency of violating the window constraints is low.
   (3) Penalty of violation of vehicle capacity: M minutes. In the experiment, vehicle capacity is not to be violated, thus the penalty cost is a big number.

Through the parameter setting, the basic experiments are simulated 5 times (T-1 – T-5), and the results are summarized in Table 2. In these basic experiments 5 or 6 vehicles are deployed to serve the customers and the average travel time is about 194.82 minutes. The total travel times range from 169.48 to 217.44 minutes and the difference is about 48 minutes. The computational time is about 0.2 CPU seconds.
Table 2 Performance indexes for the basic experiments

<table>
<thead>
<tr>
<th>Cases</th>
<th>T-1</th>
<th>T-2</th>
<th>T-3</th>
<th>T-4</th>
<th>T-5</th>
<th>Average</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total travel time (in minutes)</td>
<td>217.44</td>
<td>209.36</td>
<td>169.48</td>
<td>196.98</td>
<td>180.85</td>
<td>194.82</td>
<td>19.78</td>
</tr>
<tr>
<td>Total number of CVOs</td>
<td>6</td>
<td>6</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5.6</td>
<td>0.55</td>
</tr>
<tr>
<td>Total violation of vehicle capacity</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total violation of time-window constraints (in minutes)</td>
<td>26.28</td>
<td>18.46</td>
<td>18.22</td>
<td>25.84</td>
<td>16.42</td>
<td>21.04</td>
<td>4.65</td>
</tr>
<tr>
<td>CPU time (millisecond, ms)</td>
<td>225.43</td>
<td>257.11</td>
<td>215.92</td>
<td>235.57</td>
<td>184.44</td>
<td>223.69</td>
<td>26.75</td>
</tr>
</tbody>
</table>

4.3 Optimal Parameter Setting and Result Analysis

In the hybrid algorithm Tabu-GA, three parameters: candidate list, mutation rate, and elimination cycle are experimented to obtain the optimal parameter setting and compared with the basic experiments to observe the performance of individual parameters. In the following experiments, each case is simulated 5 times and the average value is illustrated in the comparison.

1. Candidate list

Different candidate lists are experimented, including 0, 100, 200, and 400, termed as C-0, C-100, C-200, C-400, and the results are summarized in Table 3. The results indicate that the candidate list affects the solution accuracy and computational time. If the list is short, it is likely that the algorithm can only find the local optimal. If the list is long, the computational time is increasing exponentially. Through the experiments, the appropriate candidate list is about 200.

Table 3 Impact of the length of candidate list

<table>
<thead>
<tr>
<th>Cases</th>
<th>C-0</th>
<th>C-100</th>
<th>C-200</th>
<th>C-400</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total travel time (in minutes)</td>
<td>182.98</td>
<td>194.82</td>
<td>171.42</td>
<td>162.91</td>
</tr>
<tr>
<td>Total number of CVOs</td>
<td>6</td>
<td>5.6</td>
<td>5.6</td>
<td>5.2</td>
</tr>
<tr>
<td>Total violation of vehicle capacity</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total violation of time-window constraints (in minutes)</td>
<td>24.38</td>
<td>21.04</td>
<td>18.27</td>
<td>14.61</td>
</tr>
<tr>
<td>CPU time (millisecond, ms)</td>
<td>6.09</td>
<td>223.69</td>
<td>394.46</td>
<td>773.61</td>
</tr>
</tbody>
</table>

2. Mutation rates

Several mutation rates are experimented, including 0%, 1%, 10%, and 100%, termed as M-0, M-1, M-10, M-100, and the results are summarized in Table 4. The mutation rate might affect solution stability. The rate indicates the ability of the algorithm to jump out from local optimal region. If the mutation rate is high, it is expected that the solution stability is low. On the other hand, if the mutation rate is low, the ability of jumping out from local solution regions is relatively weak. As shown in Table 4, although the difference is not significant, the mutation rate of 10% is appropriate.

Table 4 Impact of mutation rates

<table>
<thead>
<tr>
<th>Cases</th>
<th>M-0</th>
<th>M-1</th>
<th>M-10</th>
<th>M-100</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total travel time (in minutes)</td>
<td>176.03</td>
<td>194.82</td>
<td>172.06</td>
<td>178.33</td>
</tr>
<tr>
<td>Total number of CVOs</td>
<td>5.6</td>
<td>5.6</td>
<td>5.6</td>
<td>5</td>
</tr>
<tr>
<td>Total violation of vehicle capacity</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total violation of time-window constraints (in minutes)</td>
<td>26.06</td>
<td>21.04</td>
<td>23.92</td>
<td>22.36</td>
</tr>
<tr>
<td>CPU time (millisecond, ms)</td>
<td>207.48</td>
<td>223.69</td>
<td>213.43</td>
<td>216.97</td>
</tr>
</tbody>
</table>

3. Elimination cycle
Several elimination cycles are experimented, including 1, 10, 100, and infinity, termed as E-1, E-10, E-100, E-i, and the results are summarized in Table 5. The elimination cycle affect the evolution of chromosomes. If the elimination cycle is too short, chromosomes do not have enough chance to evolve; on the contrary, it is much more difficult to remove inferior chromosomes with longer elimination cycles. The elimination cycle is chosen to be 100.

<table>
<thead>
<tr>
<th>Cases</th>
<th>E-1</th>
<th>E-10</th>
<th>E-100</th>
<th>E-i</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total travel time (in minutes)</td>
<td>172.56</td>
<td>194.82</td>
<td>174.07</td>
<td>209.83</td>
</tr>
<tr>
<td>Total number of CVOs</td>
<td>5.4</td>
<td>5.6</td>
<td>5.2</td>
<td>5.4</td>
</tr>
<tr>
<td>Total violation of vehicle capacity</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total violation of time-window constraints (in minutes)</td>
<td>17.75</td>
<td>21.04</td>
<td>21.32</td>
<td>31.66</td>
</tr>
<tr>
<td>CPU time (millisecond, ms)</td>
<td>32.78</td>
<td>223.69</td>
<td>250.67</td>
<td>298.27</td>
</tr>
</tbody>
</table>

Based on the optimal parameter setting, the basic experiments are then repeated and the results are summarized in Table 6. Compared with Table 2, performance indexes are improved over the previous experiments. The total travel time drops from 194.82 to 172.09 minutes, about 11% of reduction. Another observation is the reduction of standard deviation, and the results show the stability of the solution process under optimal parameter setting. Although the computational times slightly increase to about 0.5 CPU second, the algorithm is still very efficient.

<table>
<thead>
<tr>
<th>Cases</th>
<th>F-1</th>
<th>F-2</th>
<th>F-3</th>
<th>F-4</th>
<th>F-5</th>
<th>average</th>
<th>Standard deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total travel time (in minutes)</td>
<td>180.19</td>
<td>184.7</td>
<td>160.82</td>
<td>166.6</td>
<td>168.16</td>
<td>172.09</td>
<td>9.97</td>
</tr>
<tr>
<td>Total number of CVOs</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>6</td>
<td>5</td>
<td>5.2</td>
<td>0.45</td>
</tr>
<tr>
<td>Total violation of vehicle capacity</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total violation of time-window constraints (in minutes)</td>
<td>15.5</td>
<td>14.68</td>
<td>13.84</td>
<td>15.39</td>
<td>20.93</td>
<td>16.07</td>
<td>2.80</td>
</tr>
<tr>
<td>CPU time (millisecond, ms)</td>
<td>542.00</td>
<td>565.93</td>
<td>520.79</td>
<td>569.32</td>
<td>571.54</td>
<td>553.92</td>
<td>21.97</td>
</tr>
</tbody>
</table>

### 5. CONCLUDING REMARKS

In this paper, the problem of fleet management is explored in two parts: off-line planning and on-line rerouting. The off-line stage is accomplished through the time-dependent VRP formulation. The real-time route updating algorithm is developed through the hybrid algorithm Tabu-GA to consider real-time requests as well as variations of link travel times. Through the experiments, the time-dependent VRP formulation provides good solutions. In the hybrid algorithm, the candidate list and elimination cycle are two major parameters. The system could achieve better performance when the length of candidate list is 200 and the elimination cycle is 100. However, the formulation is unable to solve for large demands within a reasonable time range. Although the heuristic approaches lack of theoretical properties, the hybrid approach solves on-line VRP efficiently.

The core of the framework, DynaTAIWAN, provides traffic simulation and assignment capabilities under mixed traffic flow conditions; however, only real-time current traffic condition is obtained. Possible predicted traffic conditions could possible to enhance the
benefits from real-time route updating. The framework provides a practical tool for the evaluation of vehicle routing strategies under real-time information. This capability is necessary to evaluate phenomena where time-variation is essential, including dynamic fleet management and real-time information systems.

The proposed hybrid algorithm shows its flexibility and efficiency to a variety of assumptions on Tabu and GA parameters; however, several future developments could be identified: (1) the proposed hybrid algorithm can be compared with either tabu search algorithm or GA algorithm to measure the algorithm performance, in terms of efficiency and accuracy; (2) the algorithm could be examined empirically.

ACKNOWLEDGEMENT

This paper is based on work partially supported by National Science Council, Taiwan, R.O.C. Of course, the authors are solely responsible for the contents of this paper.

REFERENCES


