Vehicle routing problem with time windows considering overtime and outsourcing vehicles

IlKyeong Moon\textsuperscript{a}, Jeong-Hun Lee\textsuperscript{b,}\textsuperscript{*}, June Seong\textsuperscript{c}

\textsuperscript{a}Department of Industrial Engineering, Pusan National University, Busan 609-735, South Korea
\textsuperscript{b}Postal & Logistics Technology Research Department, Electronics and Telecommunications Research Institute, Daejeon 305-700, South Korea
\textsuperscript{c}Overseas Business Planning Team, Sejong Industrial Co., Ltd., Ulsan 683-360, South Korea

\textbf{ARTICLE INFO}

Keywords: Vehicle routing, Outsourcing vehicle, Genetic algorithm, Simulated annealing, Decision support system

\textbf{ABSTRACT}

The vehicle routing problem with time windows (VRPTW) is an important problem in third-party logistics and supply chain management. We extend the VRPTW to the VRPTWOV, which allows overtime for drivers and the possibility of using outsourced vehicles. This problem can be applied to third-party logistics companies for managing central distributor-local distributors, local distributor-retailers (or customers), and manufacturers. We developed a mixed integer programming model, a genetic algorithm (GA), and a hybrid algorithm based on simulated annealing. Computational results demonstrate the efficiency of the developed algorithms. We also develop a decision support system for the VRPTWOV that is equipped with a vehicle route rescheduling function for realistic situations based on the GA.

\textcopyright{} 2012 Elsevier Ltd. All rights reserved.

1. Introduction

Vehicle routing problems are important and well-known combinatorial optimization problems occurring in many transport logistics and distribution systems of considerable economic significance. The vehicle routing problem with time windows (VRPTW) has been widely studied in the literature, mainly because of related real-world logistics and transportation problems (Liu & Shen, 1999; Nguyen & Kim, 2010; Zhang & Peng, 2012). Normally, the objective of VRPTW is to minimize the total distance traveled by a vehicle from a single location (the depot) to a set of geographically scattered points (the customers), but it is also common to minimize route costs. The routes must be designed in such a way that each point is visited only once by exactly one vehicle within a given time interval (time windows). All the routes start and end at the depot, and the total demand for all the points on one particular route must not exceed the capacity of the vehicle (Solomon, 1987).

The VRPTW has many real-life applications, especially in third-party logistics (3PL) (Chung & Park, 2004; Custódio & Oliveira, 2006; Karaoğlan & Altıparmak, 2011; Lee & Lee, 2005). For these problems, scalable methods capable of producing high-quality results in a limited time, even with several hundreds of customers, are particularly important. According to Lenstra and Kan (1981), the VRPTW belongs to the class of NP-hard combinatorial optimization problems. Although optimal solutions can be obtained using exact methods, the computation time required to solve the VRPTW to optimality is prohibitive (Desrochers, Johnson, Savelsbergh, & Stougie, 1999). Recently, some research has been carried out relating to the exact methods. (Chabrier, 2006; Jepsen, Petersen, Spoorendonk, & Pisinger, 2008; Kallehauge, Larsen, & Madsen, 2006). However, since heuristic methods often produce near-optimal solutions in a reasonable amount of computational time, most of the research has focused on the design of heuristics and meta-heuristics (Alabas-Uslu & Dengiz, 2011; Badeau, Guertin, Gendreau, Potvin, & Taillard, 1997; Homberger & Gehring, 2005; Ting & Huang, 2005).

In particular, the VRPTW is still "much more difficult" to solve than the VRP (Solomon & Desrosiers, 1988). Hence, heuristic procedures are suggested for larger problem instances of the VRPTW. In the past, fairly good results have been achieved for the VRPTW with meta-heuristics (Lee, Son, & Ou, 2001; Bräysy & Gendreau, 2005a, 2005b; Bouthillier & Crainic, 2005; Bräysy, Pornkula, Dullaert, Repoussis, & Tarantilis, 2009; Mester & Bräysy, 2007). In recent years, some researchers considered outsourcing decisions for vehicles or workers (Seong & Moon, 2006). Zäpfel and Bögl (2008) presented a local logistic company that provides transportation service for empty and laden containers. Due to the limited capacity of its own fleet of vehicles, they considered outsourcing for some orders to other smaller local transportation companies and presented a vehicle capacity planning system (VCPS) using the tabu search (TS). Lee, Tan, Ou, and Chew (2003) proposed a hybrid meta-heuristic combined with a construction heuristic for short-range weekly planning on the part of postal companies in consideration of variable vehicle capacities and

* Corresponding author. Tel.: +82 42 860 5222; fax: +82 42 860 6508.
E-mail addresses: ilkmoon@pusan.ac.kr (I. Moon), jhunlee@etri.re.kr (J.-H. Lee), juneeseong@sijku.co.kr (J. Seong).

0957-4174/$ - see front matter \textcopyright{} 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.05.081
decision support system that can be used for third party logistics and including outsourcing decisions for tours and drivers. Repoussis, Paraskevopoulos, Zobolas, Tarantilis, and Ioannou (2009) presented a web-based decision support system (DSS) that enables schedulers to tackle reverse supply chain management problems interactively for the efficient and effective management of waste lube oils collection and recycling operations using shortest path and sophisticated hybrid metaheuristic algorithms. While the VRPTW has been widely studied in the literature, outsourcing vehicles and overtime (i.e., regular and overtime labor cost) have not been considered simultaneously.

Therefore, this paper presents a single depot VRPTW that considers outsourcing vehicles and overtime (VRPTWOV). Owing to the limited number of vehicles and reductions in the relevant costs, third party logistics companies often need outsourcing vehicles. In such a case, the managers must consider a tradeoff between the overtime and outsourcing costs. The main objectives of this paper are to develop a mixed integer programming, a genetic algorithm (GA), and a hybrid algorithm based on simulated annealing for solving the VRPTWOV. Moreover, we develop a decision support system that can be used for third party logistics companies.

The rest of this paper is organized as follows. In Section 2, we develop a mixed integer programming model for the VRPTWOV. In Section 3, we develop a GA and a hybrid algorithm based on simulated annealing for the VRPTWOV. We present computational experiments for the developed algorithms in Section 4. In Section 5, we present a decision support system based on the developed GA. Finally, the conclusions are presented.

2. Mixed integer programming model

Given the previous problem statement, an initial approach used for solving the problem is to formulate the problem as a mixed integer programming. The model used for solving the problem is explained below.

Notations

(1) Parameters

- \( CT_k \): travel cost for vehicle \( k \) per unit time
- \( CR_k \): regular labor cost for vehicle \( k \) per unit time
- \( CO_k \): overtime labor cost for vehicle \( k \) per unit time
- \( CF_k \): fixed (or outsourcing) cost for vehicle \( k \)
- \( t_{ij} \): travel time between node \( i \) and \( j \)
- \( TR_k \): regular time(without overtime) of vehicle \( k \)
- \( s_i \): service time(loading/unloading time) at customer \( i \)
- \( e_i \): earliest time at customer \( i \)
- \( l_i \): latest time at customer \( i \)
- \( r_k \): maximum route time allowed for vehicle \( k \)
- \( m_i \): demand at customer \( i \)
- \( q_k \): capacity of vehicle \( k \)
- \( K \): total number of vehicles
- \( N \): total number of customers
- \( M \): Big M (a large number)

(2) Decision variables

- \( T_{0k} \): arrival time of vehicle \( k \) at depot
- \( TO_k \): overtime of vehicle \( k \)
- \( T_i \): arrival time at customer \( i \)
- \( w_i \): waiting time at customer \( i \)
- \( NO_k \): Max \( \{ TR_k - T_{0k}, 0 \} \) of vehicle \( k \)
- \( x_{ijk} \): \[
\begin{cases}
1, & \text{if the vehicle } k \text{ travels directly from } i \text{ to } j \\
0, & \text{otherwise}
\end{cases}
\]
where \( i, j, k \in \{0, \ldots, N\} \) is 0 at the origin or depot.
- \( Y_k \): \[
\begin{cases}
1, & \text{if the vehicle } k \text{ is used} \\
0, & \text{otherwise}
\end{cases}
\]

Objective function:

Minimize \[
\sum_{i=0}^{N} \sum_{j=0}^{N} \sum_{k=1}^{K} CT_{ij} x_{ijk} + \sum_{k=1}^{K} CR_k (T_{0k} - TO_k) + \sum_{k=1}^{K} CO_k TO_k + \sum_{k=1}^{K} CF_k Y_k
\]

Subject to

\[
\begin{align*}
\sum_{j=0}^{N} x_{ijk} & \leq 1, \quad \text{for } k \in \{1, \ldots, K\} \\
\sum_{i=0}^{N} x_{ijk} & = \sum_{i=0}^{N} x_{ijk}, \quad \text{for } i \in \{0, \ldots, N\}, \ k \in \{1, \ldots, K\}, \ \text{and } i \neq j \\
\sum_{k=1}^{K} m_i \sum_{j=0}^{N} x_{ijk} & \leq q_k, \quad \text{for } k \in \{1, \ldots, K\} \\
T_j & \geq T_i + s_i + w_i + t_0 + M (\sum_{k=1}^{K} x_{ijk} - 1), \\
& \quad \text{for } i \in \{0, \ldots, N\}, \ j \in \{1, \ldots, N\}, \ \text{and } i \neq j \\
r_k & \geq T_{0k} \geq T_i + s_i + w_i + t_0 + M (x_{ijk} - 1), \\
& \quad \text{for } i \in \{1, \ldots, N\}, \ k \in \{1, \ldots, K\} \\
e_i & \leq T_i + w_i \leq l_i, \quad \text{for } i \in \{1, \ldots, N\} \\
T_0 & = 0 = w_0 \\
NO_k - NO & = TO_k - TR_k, \quad \text{for } k \in \{1, \ldots, K\} \\
Y_k & = \sum_{j=1}^{N} x_{ijk}, \quad \text{for } k \in \{1, \ldots, K\}
\end{align*}
\]

Eq. (1) is the objective function of the problem. It seeks to minimize the sum of the total travel cost, total labor cost for regular time, total overtime labor cost, and total fixed cost for vehicles. Constraint (2) defines that every vehicle starts only once at the depot. Constraint (3) ensures that if vehicle \( k \) visits a customer, it has to leave the customer again. Constraint (4) defines that every customer can be visited only once by one vehicle. Constraint (5) is the capacity constraint of the vehicle. Constraints (6)–(9) ensure compatible arrival times and define the time windows of the vehicle at a customer. Constraint (10) defines the overtime of each vehicle. In this constraint, \( NO_k \) guarantees that if overtime occurs, \( TO_k \) has some value. Otherwise, \( TO_k = 0 \). Constraint (11) defines that if the vehicle starts at the depot, \( Y_k = 1 \). Otherwise, \( Y_k = 0 \) for a fixed cost.

3. Metaheuristics

Although optimal solutions can be obtained using exact methods, the computation time required to solve the VRPTWOV is prohibitive. Solving realistic scale VRPTWOV using the MIP (Mixed Integer Programming) model provided in the earlier section has many restrictions. Therefore, a GA and a hybrid algorithm for the VRPTWOV are presented in this section.
3.1. Genetic algorithm (GA)

GAs were invented and developed initially by Holland and his associates at the University of Michigan in the 1960s and 1970s to mimic some of the processes observed in natural selection (Goldberg, 1989). GAs, which have been widely used in various areas for three decades, are stochastic search algorithms based on the mechanism of natural selection and natural genetics. Many research papers and dissertations have established the validity of GA approach in optimization problems and application problems. A GA, in contrast to conventional search techniques, begins with an initial set of random solutions called population. Each individual in the population is called a chromosome that represents a solution to the problem. The chromosomes evolve through successive iterations called generations. During each generation, the chromosomes are evaluated by taking some measures of fitness. The offspring is formed by merging two chromosomes from the current generation using the crossover operator and/or modifying a chromosome using the mutation operator. A new generation is selected according to the fitness values of the parents and offspring, and then poor chromosomes are weeded out in order to maintain a constant population size (Mitchell, 1998). It is generally accepted that any GA must have four basic components to solve a problem: further, they must have different characteristics depending on the problem under study. We explain their overall strategies including the chromosome style as follows:

- Representation and initialization
- Objective and fitness function
- Reproduction, crossover, and mutation
- Terminating condition and parameters

3.1.1. Representation and initialization

In this paper, we use two kinds of chromosomes. The first chromosome represents the sequence for customers. The second chromosome represents the assignment of vehicles. The length of the sequence chromosome for customers equals the number of all customers. Each gene represents sequences for customer visitation. The assignment chromosome for vehicles assigns the number of customers by a randomly evaluated length of genes that is calculated as “the genes of each vehicle / sum of all vehicles’ genes * total number of customers”. Table 1 shows an example of the sequence chromosome for seven customers. The sequence for the customers to each vehicle by an assignment calculation (i.e., one customer is assigned to the first vehicle, three customers are assigned to the second vehicle, three customers are assigned to the third vehicle). The VRPTWOV is highly dependent on the initial solutions. Therefore, we create a feasible initial solution for large size problems using a fast greedy heuristic. This heuristic consists of a sorting by average of the time windows of each customer and the assignment of vehicles by turns (Table 3).

3.1.2. Objective and fitness function

A fitness function is computed for each chromosome in the population. The objective is to minimize the sum of the travel cost, regular labor cost, overtime cost, and fixed cost for vehicles, which is the same objective function, Eq. (1), as that of the mixed integer program in Section 2.

\[ \text{eval}(x) = f(x), \]

where \( f(x) \) is the objective function and \( x \) represents a chromosome.

3.1.3. Reproduction, crossover, and mutation

A simple GA that yields good results in the case of many practical problems is composed of three operators: reproduction, crossover, and mutation. In this paper, parents are chosen based on a rank-based mechanism. All solutions are ranked according to their fitness. The amount of pheromone deposited is then weighted for each solution, such that the solutions with better fitness deposit more pheromone than the solutions with worse fitness. Instead of a GA system where a parent’s chance of selection for reproduction is directly proportional to its fitness, a ranking approach offers a smoother selection probability curve. This prevents good organisms from completely dominating the evolution from an early point. We produce offsprings for sequence chromosomes using

<table>
<thead>
<tr>
<th>Gene</th>
<th>7</th>
<th>4</th>
<th>6</th>
<th>5</th>
<th>2</th>
<th>3</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
</tr>
</tbody>
</table>

Table 1

An example of a sequence chromosome for customers.

<table>
<thead>
<tr>
<th>Vehicle number</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of assigned customers</td>
<td>1 ((0.1876)/7)</td>
<td>3 ((0.8943)/7)</td>
<td>3 ((0.9577)/7)</td>
</tr>
</tbody>
</table>

Table 2

An example of an assignment chromosome for vehicles.

<table>
<thead>
<tr>
<th>Customer</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average of the time windows</td>
<td>138.5</td>
<td>82.5</td>
<td>120.5</td>
<td>134</td>
<td>93.5</td>
<td>103.5</td>
<td>85.5</td>
</tr>
<tr>
<td>Order</td>
<td>7</td>
<td>1</td>
<td>5</td>
<td>6</td>
<td>3</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Assigned vehicle number</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 3

An example of greedy heuristic for creating a feasible initial solution.

1. Select the substrings

Parent 1

\[
\begin{array}{cccccc}
7 & 4 & 6 & 5 & 2 & 3 \\
1 & 2 & 3 & 4 & 5 & 7 & 6
\end{array}
\]

Parent 2

2. Exchange substrings

Proto-offspring 1

\[
\begin{array}{cccccc}
7 & 4 & 3 & 4 & 5 & 3 & 1 \\
1 & 2 & 6 & 5 & 2 & 7 & 6
\end{array}
\]

Proto-offspring 2

3. Mapping relationship

\[ 3 \leftrightarrow 6, 4 \leftrightarrow 5 \leftrightarrow 2 \]

4. Legalize the offspring

Offspring 1

\[
\begin{array}{cccccc}
7 & 2 & 3 & 4 & 5 & 6 & 1 \\
1 & 4 & 6 & 5 & 2 & 7 & 3
\end{array}
\]

Offspring 2

Fig. 1. An example of PMX procedure.
partially matched crossover (PMX) because each gene represents the sequences for customer visitation. PMX may be viewed as a crossover of permutations that guarantees that all positions are found exactly once in each offspring, i.e., both offspring receive a full complement of genes, followed by the corresponding filling in of alleles from their parents. PMX proceeds as follows: (1) Select a substring uniformly in two parents at random. (2) Exchange these two substrings to produce proto-offspring. (3) Determine the mapping relationship according to these two substrings. (4) Legalize proto-offspring with the mapping relationship (Fig. 1).

We use one-point crossover for the assignment chromosome for the vehicle. A general mutation operator is used that generates random numbers and replaces each gene.

3.1.4. Terminating condition and parameters

In order to choose the appropriate parameter values in the GAs to minimize the sum of cost, we employed the above-mentioned crossover and mutation operator with a crossover rate of 0.5 and a mutation rate of 0.1 and the size of the population is 100. The terminating condition was to stop the algorithm when 200,000 (in the case of 50 customers) generations were reached or when the best individual does not become better than 0.01% during 1000 generations. The values of the parameters were decided after a pilot test.

3.2. Hybrid algorithm

Simulated annealing (SA) transfers a physical process—in which a solid is first melted at a high temperature and then cooled very slowly to obtain a perfect lattice structure—to a local search algorithm for combinatorial optimization problems (Kirkpatrick, Gelatt, & Vecchi, 1983). It can avoid a local optimum by accepting worse solutions with some probability (Dréo,PETROWSKI, SIarry, & Taillard, 2006). SA is very powerful in solving complicated problems such as combinatorial problems. However, due to the random processes used to search for the minimum energy state, the convergence speed of SA is very slow (Hwang & He, 2006). Therefore, a hybrid SA is developed to avoid the slow speed of the SA and premature convergence of the GA, and improve the hill-climbing ability of the GA. As mentioned in Section 3.1, the VRPTWOW is highly dependent on the initial solutions. We create a feasible initial solution using a greedy heuristic. This hybrid algorithm can retain the advantages and avoid the disadvantages of both search algorithms. Owing to the application of SA, this algorithm has a better fine-turning ability to search for global optimal solutions. In a hybrid algorithm, the GA and SA are cooperatively used. In this method, the convergence of the GA is improved by introducing the probability of SA as the criterion for the acceptance of a new trial solution. The hybrid algorithm for solving this problem consists of four components. The detailed components are as follows:

- Initial solution generation
- Neighborhood generation (perturb)
- Evaluation and update (cooling schedule)
- Terminating condition and parameters

3.2.1. Initial solution generation

Since the VRPTWOW is highly dependent on the initial solutions, a feasible initial solution is created using a greedy heuristic. This heuristic is very suitable for small-sized problems. However, in large-sized problems (e.g., more than 50 customers), sometimes it cannot obtain a feasible solution because of the more tight time windows. Therefore, a modified objective function including a penalty function is used.

eval(x) = f(x) + p(x),

p(x) = 0, if x is feasible,

p(x) > 0, otherwise,

where f(x) is the objective function, p(x) is the penalty function, and x represents a chromosome.

| Table 4 | Customer demands (unit) and time windows (minute) data. |
|---|---|---|---|---|---|---|---|---|
| Customer | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Demand | 10 | 7 | 7 | 26 | 3 | 5 | 9 | 16 |
| Earliest time | 73 | 18 | 76 | 20 | 49 | 36 | 50 | 47 |
| Latest time | 204 | 147 | 165 | 167 | 158 | 135 | 149 | 156 |
| Service time | 10 | 10 | 10 | 10 | 10 | 10 | 10 | 10 |

| Table 5 | Distance matrix of vehicle operation for each customer (minute). |
|---|---|---|---|---|---|---|---|---|
| Depot | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| 0 | 15 | 18 | 22 | 25 | 21 | 11 | 21 | 26 | 32 |
| 1 | 15 | 33 | 30 | 34 | 20 | 24 | 16 | 36 | 47 |
| 2 | 22 | 15 | 34 | 0 | 25 | 43 | 34 | 45 | 15 |
| 3 | 25 | 32 | 20 | 25 | 0 | 41 | 32 | 46 | 51 | 40 |
| 4 | 21 | 32 | 24 | 43 | 41 | 0 | 10 | 21 | 14 | 50 |
| 5 | 11 | 25 | 16 | 34 | 32 | 10 | 0 | 21 | 20 | 42 |
| 6 | 21 | 21 | 36 | 35 | 46 | 21 | 21 | 0 | 12 | 36 |
| 7 | 26 | 32 | 36 | 45 | 51 | 42 | 20 | 12 | 0 | 48 |
| 8 | 32 | 18 | 47 | 15 | 40 | 50 | 42 | 36 | 48 | 0 |
of the best solution), or if the value of \( C(y) \) is better than the cost of the best solution and the cost of the neighboring solution.

### 3.2.3. Evaluation and update

In the GA, crossover and mutation are used by applying the same method given in Section 3.1.3.

### 3.2.2. Neighborhood generation

It is very difficult and slow to create a neighborhood generation using a simple heuristic. Therefore, we create a next generation solution using the GA since the search speed of the GA is very fast. In the GA, crossover and mutation are used by applying the same method given in Section 3.1.3.

### 3.2.4. Terminating condition and parameters

For the creation of a high quality solution, we choose the appropriate parameters using the preliminary experiments. We used an initial temperature of 100 \( (T_{\text{initial}}) \), a final temperature of 1 \( (T_{\text{last}}) \), number of iterations \( (I) \) in the SA equals to 200,000, a crossover rate of 0.5, and a mutation rate of 0.1 in the GA. The termination condition was to stop the algorithm when the number of iterations reached 200,000 and the temperature became less than \( T_{\text{last}} \). The procedure of the hybrid algorithm is shown in Fig. 2.

### 4. Computational experiments

The mixed integer programming model was implemented and solved using LINDO and the heuristic algorithms (GA and Hybrid algorithm) were implemented using Microsoft Office Excel 2003

---

### Table 6

<table>
<thead>
<tr>
<th>Number of customers</th>
<th>Vehicle capacity</th>
<th>Mixed integer programming</th>
<th>Genetic algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Computation time(^a)</td>
<td>Objective function ($)</td>
</tr>
<tr>
<td>5</td>
<td>50</td>
<td>5 s</td>
<td>238.3</td>
</tr>
<tr>
<td>7</td>
<td>50</td>
<td>7 min</td>
<td>246.0</td>
</tr>
<tr>
<td>9</td>
<td>60</td>
<td>14 h</td>
<td>259.7</td>
</tr>
</tbody>
</table>

\(^a\) Average of 10 evaluations.

### Table 7

<table>
<thead>
<tr>
<th>Customer demands and time windows (number of customers = 25).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customer</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
<tr>
<td>3</td>
</tr>
<tr>
<td>4</td>
</tr>
<tr>
<td>5</td>
</tr>
<tr>
<td>6</td>
</tr>
<tr>
<td>7</td>
</tr>
<tr>
<td>8</td>
</tr>
<tr>
<td>9</td>
</tr>
</tbody>
</table>

### Table 8

<table>
<thead>
<tr>
<th>Distance of vehicle operating for each customer (number of customers = 25).</th>
</tr>
</thead>
<tbody>
<tr>
<td>Depot</td>
</tr>
</tbody>
</table>

---
and VBA. This model needs an unacceptable amount of computational time for finding an optimal solution. We first compared the mixed integer programming and GA for small size problems in order to test their validity. They used a modified R112 example in the R1 type problem from the benchmark problems of Solomon (1987). Solomon’s problems were modified by changing the maximum travel time. The regular labor time, the fixed cost for basic vehicle, the fixed cost for outsourcing vehicle, the travel cost, the regular labor cost, and the overtime labor cost have been added. The example for comparison is as follows:

- Number of depots: 1
- Number of customers: 5, 7, 9
- Number of vehicles: 3 (2 basic vehicles, 1 outsourcing vehicle)
- Vehicle capacity: 50 (number of customers: 5, 7), 60 (number of customers: 9)
- Maximum travel time: 230 min
- Regular labor time: 120 min
- Fixed cost for basic vehicle: $100/vehicle

Table 9

<table>
<thead>
<tr>
<th>Problem instance</th>
<th>Genetic algorithm</th>
<th>Hybrid algorithm</th>
<th>Remark (outperform)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Averagea</td>
<td>Min</td>
<td>Max</td>
</tr>
<tr>
<td>RC201</td>
<td>5144.25</td>
<td>4821.26</td>
<td>5372.60</td>
</tr>
<tr>
<td>RC202</td>
<td>4624.95</td>
<td>4464.03</td>
<td>4846.89</td>
</tr>
<tr>
<td>RC203</td>
<td>4586.31</td>
<td>4251.82</td>
<td>4769.23</td>
</tr>
<tr>
<td>RC204</td>
<td>3159.40</td>
<td>2486.94</td>
<td>3505.32</td>
</tr>
<tr>
<td>RC205</td>
<td>4799.64</td>
<td>4681.70</td>
<td>4913.49</td>
</tr>
<tr>
<td>RC206</td>
<td>4449.14</td>
<td>4267.02</td>
<td>4620.76</td>
</tr>
<tr>
<td>RC207</td>
<td>4105.44</td>
<td>3966.63</td>
<td>4308.10</td>
</tr>
<tr>
<td>RC208</td>
<td>3430.05</td>
<td>3073.85</td>
<td>3681.49</td>
</tr>
<tr>
<td>C201</td>
<td>9747.55</td>
<td>9628.52</td>
<td>9929.39</td>
</tr>
<tr>
<td>C202</td>
<td>9439.87</td>
<td>9347.84</td>
<td>9559.80</td>
</tr>
<tr>
<td>C203</td>
<td>9121.16</td>
<td>8482.86</td>
<td>9381.28</td>
</tr>
<tr>
<td>C204</td>
<td>6754.85</td>
<td>5901.01</td>
<td>7758.45</td>
</tr>
<tr>
<td>C205</td>
<td>9376.35</td>
<td>9155.65</td>
<td>9484.13</td>
</tr>
<tr>
<td>C206</td>
<td>9150.67</td>
<td>9052.05</td>
<td>9207.14</td>
</tr>
<tr>
<td>C207</td>
<td>8440.32</td>
<td>8244.43</td>
<td>8500.87</td>
</tr>
<tr>
<td>C208</td>
<td>8813.16</td>
<td>8604.29</td>
<td>8905.07</td>
</tr>
</tbody>
</table>

a Average of 10 evaluations.
Tables 4 and 5 show the parameters for the customers. Table 6 shows a comparison of the results obtained from the mixed integer programming and GA. Each objective function has the same value. Due to the characteristics of the NP-hard problem, the computational time increases exponentially with the number of customer increases.

We experiment with a larger size problem having 25 customers. Tables 7 and 8 show the parameters for the customers. The detailed data are as follows:

- Number of depots: 1
- Number of customers: 25
- Number of vehicles: 3 (2 basic vehicles, 1 outsourcing vehicle)
- Vehicle capacity: 700
- Maximum travel time: 3390 minutes
- Regular labor time: 2800 minutes
- Cost parameters are the same as in the previous example

The route of this problem is determined using the developed GA. The results of this problem are as follows:

- Overtime labor cost: $0.2/min

Fig. 5. Average total cost for the C2 type instances.

Fig. 6. Process flow of the decision support system.
Total cost: $1244.0
A. Travel cost: $314.5
B. Fixed cost for vehicles: $350.0
C. Regular time labor cost: $570.6
D. Overtime labor cost: $8.9

- Total cost: $1244.0
  A. Travel cost: $314.5
  B. Fixed cost for vehicles: $350.0
  C. Regular time labor cost: $570.6
  D. Overtime labor cost: $8.9
- Run sequence (0: depot)
  A. Vehicle No. 1: 0-2-1-5-22-20-0
  B. Vehicle No. 2: 0-24-6-23-18-16-14-12-15-19-17-25-0
  C. Vehicle No. 3: 0-4-3-7-13-9-11-10-8-21-0
- Overtime

A. Vehicle No. 3: 44.5 min

The vehicle route is shown in Fig. 3.

For the experimental investigation, the modified 50-customer problem instances of Solomon (1987) were used. The benchmark set contains six different subsets called R1, R2, RC1, RC2, C1, and C2. The locations of the customers are uniformly distributed in R1 and R2, and clustered in C1 and C2. For the groups RC1 and RC2, the clustered and random distributions are mixed. For instances of Type 2, the vehicle capacity and time windows are less...
constraining. Hence, more customers can be served by one vehicle, and fewer vehicles are needed. We used some modified RC2 type problems and C2 type problems because their problem was more complicated than the VRPTW. The computational experiments were conducted on a 3.0 GHz PC with 512 MB RAM on the Microsoft Windows XP operating system.

Table 9 shows the results of the GA and hybrid algorithm for the benchmark problems. It took about 10 minutes for both the GA and hybrid algorithm found a solution. The hybrid algorithm outperforms the GA in terms of the average total cost of the RC2 type problems. However, in the C2 type problems, the GA is slightly better than the hybrid algorithm. Consequently, the hybrid
algorithm is suitable for the more random location customers and smaller time windows, and the GA is suitable for the clustered customers and bigger time windows. Figs. 4 and 5 illustrate the average total cost of the GA and hybrid algorithm for the benchmark instances, respectively.

5. Decision support system

We develop a decision support system based on the genetic algorithm developed in Section 3. It is developed by Microsoft® Visual Basic 6.0 using the SQL Server 2000. The application provides
vehicle routing selection processes and scheduling optimization by considering the travel time, the capability of vehicles, and the number of possible vehicles. It provides the information of the vehicles and items by utilizing GPS and RFID technologies. It focuses on the user-oriented design for easy applicability to realistic business environments. Fig. 6 shows the process flow of the decision support system.

We defined many factors of the input data for searching the optimal routes in VRPTWV. The input data factors are basic vehicle information, outsourcing vehicle information, customer information, and delivery/pick-up quantity. Fig. 7 shows the optimal vehicle routes and vehicle operation sequences for two basic vehicles and one outsourcing vehicle for the problem with one depot and twenty-five customers. This system provides arrival/start/waiting time by taking into account the time windows. The total cost information on the screen considered the number of vehicles, the total tour time of the selected vehicles, the quantities of the pick-up/delivery, the regular labour cost, and the overtime costs. The vehicle routing sequences are printed graphically on the screen.

The user can determine the customer location and routes for each vehicle on the map. Detailed information of the total costs, time schedule, and vehicle routing sequences are displayed on the screen. Fig. 8 shows the detailed information of one of the basic vehicles. From this screen, the user can determine detailed information of each vehicle, such as detailed routing sequences and arrival/start time for each customer.

In realistic vehicle routing, one can use additional vehicles and change the vehicle routing sequence. To solve this problem, this system provides an alternative solution module. The user can simulate by changing the number of vehicles and comparing the total cost of the alternative result with the total cost of the original result. This function enables the user to choose the better solution by comparing the routing sequence for each vehicle and the factors that influenced the total cost. Fig. 9 shows the alternative solution analysis screen.

This system provides real-time information for enhancing the customer service. Fig. 10 shows the real-time information screen. This screen provides the real-time vehicle route tracking information. Optimal vehicle routing on realistic logistic environments using GPS/RFID technologies has been applied. This process makes it possible for the user to track the location of each vehicle and determine the current customer, expected departure time from the current customer, arrival time to the next customer, and information on the vehicle. One needs to consider additional vehicles or change the run sequences because the routing sequences may be changed during the working time depending on the situation. For example, when some customers want to change the visiting time (time windows) or some vehicles cannot be used due to accidents or traffic jams, one needs to consider rescheduling. To solve this problem, this system provides a rescheduling algorithm using the GA. Figs. 11 and 12 show the rescheduling screen of the vehicle routes.

6. Conclusions

A mathematical model for the vehicle routing problem with time windows, outsourced vehicle, and overtime (VRPTWVO) has been constructed in this work. We introduced a GA and a hybrid algorithm based on simulated annealing. These algorithms can be applied to third party logistics companies for managing central distributor-local distributors and manufacturers. The computational results demonstrate the efficiency of the developed algorithms. A decision support system based on this GA has been developed. Using this system, the managers in logistics companies can determine the near-optimal route for vehicles. Additionally, rescheduling and an alternative solution analysis process can be used to solve realistic vehicle routing problems. Developing other metaheuristics (i.e., Ant Colony Optimization, Particle Swarm Optimization) for the VRPTWVO and comparing them with our hybrid algorithm might be an interesting research problem.

Acknowledgements

This work was supported by a 2-Year Research Grant of Pusan National University. This paper is an extended version of the preliminary study published in the Journal of Korean Institute of Industrial Engineers in 2006. References


