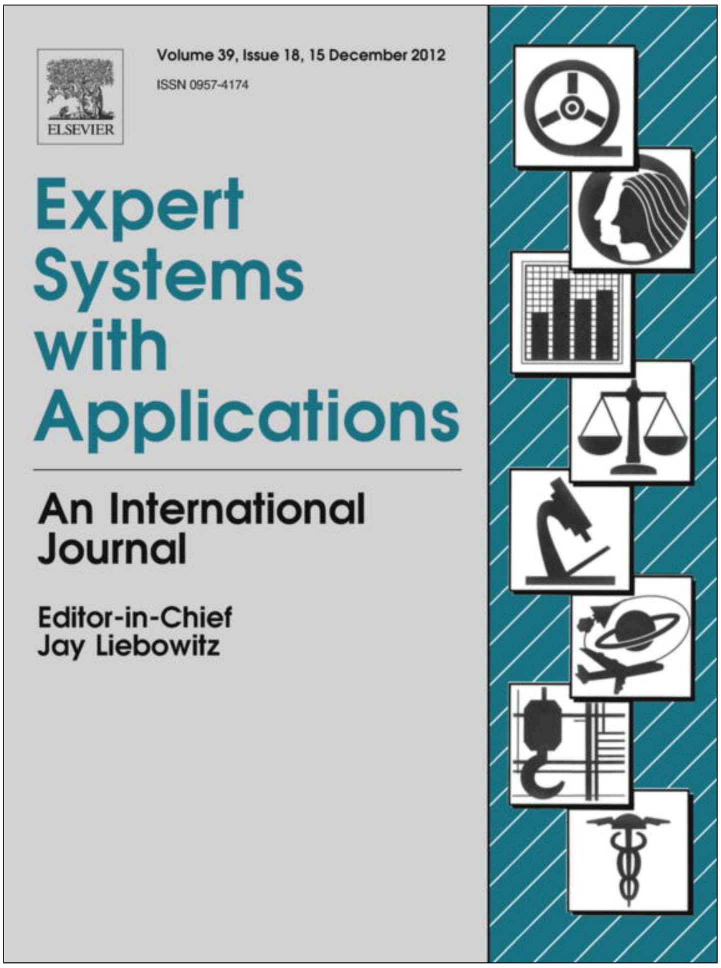
Volume 39, Issue 18, 15 December 2012
ISSN 0957-4174
ELSEVIER
Expert Systems with Applications
An International Journal
Editor-in-Chief
Jay Liebowitz

Expert Systems with Applications 39 (2012) 13202–13213

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Vehicle routing problem with time windows considering overtime and outsourcing vehicles

IlKyeong Moon [a], Jeong-Hun Lee [b,*], June Seong [c]

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

ARTICLE INFO

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

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 VRPTW with overtime and outsourcing vehicles (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. The 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.

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; Karaoglan & Altiparmak, 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, Jones, Lenstra, 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 metaheuristics (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 recent past, fairly good results have been achieved for the VRPTW with metaheuristics (Tan, Lee, Zhu, & Ou, 2001; Bräysy & Gendreau, 2005a, 2005b; Bouthillier & Crainic, 2005; Bräysy, Porkka, 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 metaheuristic 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: ikmoon@pusan.ac.kr (I. Moon), jhunlee@etri.re.kr (J.-H. Lee), juneseong@sjku.co.kr (J. Seong).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.05.081

personnel planning, 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 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}	$\left\{\begin{array}{ll}1, & \text{if the vehicle } k \text{ travels directly from } i \text{ to } j\\ 0, & \text{otherwise}\end{array}\right\}$ where $i, j \in \{0, \cdots, N\}$ is 0 at the origin or depot.
Y_k	$\left\{\begin{array}{ll}1, & \text{if the vehicle } k \text{ is used}\\ 0, & \text{otherwise}\end{array}\right\}$

Objective function:

$$\text{Minimize} \quad \sum_{i=0}^{N}\sum_{j=0,j\neq i}^{N}\sum_{k=1}^{K} CT_k t_{ij} x_{ijk} + \sum_{k=1}^{K} CR_k (T_{0k} - TO_k) + \sum_{k=1}^{K} CO_k TO_k + \sum_{k=1}^{K} CF_k Y_k \tag{1}$$

Subject to

$$\sum_{j=1}^{N} x_{0jk} \leqslant 1, \quad \text{for } k \in \{1,\ldots,K\} \tag{2}$$

$$\sum_{j=0}^{N} x_{ijk} = \sum_{j=0}^{N} x_{jik}, \quad \text{for } i \in \{0,\ldots,N\},\ k \in \{1,\ldots,K\}, \text{ and } i \neq j \tag{3}$$

$$\sum_{k=1}^{K}\sum_{i=0,i\neq j}^{N} x_{ijk} = 1, \quad \text{for } j \in \{1,\ldots,N\} \tag{4}$$

$$\sum_{i=1}^{N} m_i \sum_{j=0,i\neq j}^{N} x_{ijk} \leqslant q_k, \quad \text{for } k \in \{1,\ldots,K\} \tag{5}$$

$$T_j \geqslant T_i + s_i + w_i + t_{ij} + M\left(\sum_{k=1}^{K} x_{ijk} - 1\right), \quad \text{for } i \in \{0,\ldots,N\}, j \in \{1,\ldots,N\}, \textit{ and } i \neq j \tag{6}$$

$$r_k \geqslant T_{0k} \geqslant T_i + s_i + w_i + t_{i0} + M(x_{i0k} - 1), \quad \text{for } i \in \{1,\ldots,N\},\ k \in \{1,\ldots,K\} \tag{7}$$

$$e_i \leqslant T_i + w_i \leqslant l_i, \quad \text{for } i \in \{1,\ldots,N\} \tag{8}$$

$$T_0 = s_0 = w_0 = 0 \tag{9}$$

$$TO_k - NO_k = T_{0k} - TR_k, \quad \text{for } k \in \{1,\ldots,K\} \tag{10}$$

$$Y_k = \sum_{j=1}^{N} x_{0jk}, \quad \text{for } k \in \{1,\ldots,K\} \tag{11}$$

$$\forall x_{ijk}, \quad \forall Y_k \in \{0,1\} \tag{12}$$

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 is 0. Constraint (11) defines that if the vehicle starts at the depot, Y_k is 1. Otherwise, Y_k is 0 for a fixed cost.

3. Metaheuristics

Although optimal solutions can be obtained using exact methods, the computation time required to solve the VRPTWOV to optimality 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.

Table 1
An example of a sequence chromosome for customers.

Gene	7	4	6	5	2	3	1
Customer	1	2	3	4	5	6	7

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) $\times$ total number of customers". Table 1 shows an example of the sequence chromosome for seven customers. The sequence for the visitation is 7-5-6-2-4-3-1. Table 2 shows an example of the assignment chromosome for vehicles. This chromosome assigns 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 2
An example of an assignment chromosome for vehicles.

Gene	0.1876	0.8943	0.9577
Vehicle number	1	2	3
Number of assigned customers	1 (0.1876/ 2.0396 $\times$ 7)	3 (0.8943/ 2.0396 $\times$ 7)	3 (0.9577/ 2.0396 $\times$ 7)

Table 3
An example of greedy heuristic for creating a feasible initial solution.

Customer	1	2	3	4	5	6	7
Average of the time windows	138.5	82.5	120.5	134	93.5	103.5	85.5
Order	7	1	5	6	3	4	2
Assigned vehicle number	1	1	2	3	3	1	2

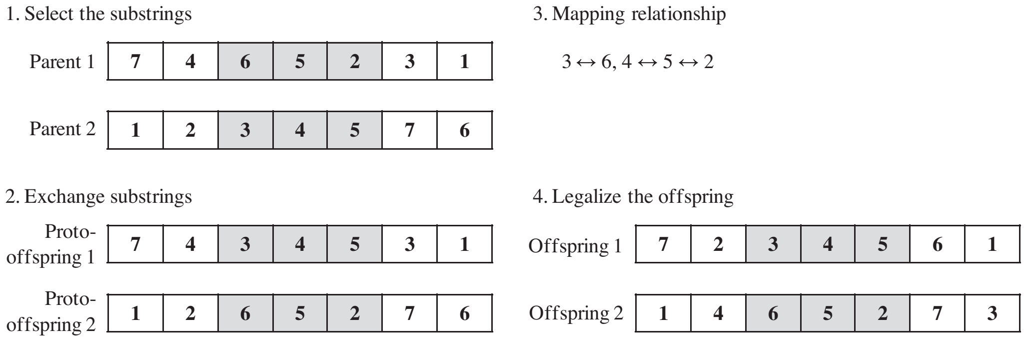

Fig. 1. An example of PMX procedure.

```
Hybrid Algorithm

Begin:
  INITIALIZE (X, Y, L)
    Set SA(X) ← Greedy Heuristic
    Repeat
      For i=1 to L do
        Y = GA(X)

          Genetic Algorithm

          Begin GA(X):
            INITIALIZE (X) ← SA(X)
            CREATE (X') {Crossover (X) & mutation (X)}
            If X' is 'feasible' Then
              Y = X'
            Else: go CREATE (X')
            End if
          End

        If (C(Y) <= C(X)) or (Exp((C(X) - C(Y))/T) > random (0,1) Then
          X=Y; {accept the movement}
          UPDATE SA(X)
        End if;
      End for;
      UPDATE (T, L)
    Until (Stop-criterion)
End
```

Fig. 2. Procedure of the hybrid algorithm.

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, Pétrowski, Siarry, & Taillard, 2006). SA is very powerful in solving complicated problems such as combinational 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 VRPTWOV 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 VRPTWOV 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.

$$\mathrm{eval}(x) = f(x) + p(x),$$

$$p(x) = 0, \ \text{if } x \text{ is feasible},$$

$$p(x) > 0, \ \text{otherwise},$$

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	13	19	26	3	5	9	16
Earliest time	73	18	76	73	20	49	36	50	47
Latest time	204	147	165	195	167	158	135	149	156
Service time	10	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	9
Depot	0	15	18	22	25	21	11	21	26	32
1	15	0	33	15	32	32	25	21	32	18
2	18	33	0	34	20	24	16	36	36	47
3	22	15	34	0	25	43	34	35	45	15
4	25	32	20	25	0	41	32	46	51	40
5	21	32	24	43	41	0	10	21	14	50
6	11	25	16	34	32	10	0	21	20	42
7	21	21	36	35	46	21	21	0	12	36
8	26	32	36	45	51	42	20	12	0	48
9	32	18	47	15	40	50	42	36	48	0

Table 6
Comparison result of mixed integer programming and genetic algorithm.

Number of customers	Vehicle capacity	Mixed integer programming			Genetic algorithm		
		Computation time[a]	Objective function ($)	Remark	Computation time[a]	Objective function ($)	Remark
5	50	5 s	238.3	Optimal	5 s	238.3	Optimal
7	50	7 min	246.0	Optimal	10 s	246.0	Optimal
9	60	14 h	259.7	Optimal	20 s	259.7	Optimal

[a] Average of 10 evaluations.

Table 7
Customer demands and time windows (number of customers = 25).

Customer	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Demand	10	30	10	10	10	20	20	20	10	10	10	20	30	10	40	40	20	20	10	10	20	20	10	10	40
Earliest time	71	20	927	1021	15	257	833	2647	2361	2551	2458	1879	2165	1786	1976	1694	2071	1502	1597	10	2648	12	1403	15	2264
Latest time	711	660	1567	1661	655	897	1473	3287	3001	3191	3098	2519	2805	2426	2616	2334	2711	2142	2237	650	3288	652	2043	655	2904
Service time	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90	90

Table 8
Distance of vehicle operating for each customer (number of customers = 25).

	Depot	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Depot	0	28	21	29	26	15	25	27	12	23	17	20	38	31	39	36	40	33	35	39	10	12	12	31	15	24
1	28	0	9	12	12	14	49	8	23	25	19	18	29	30	32	32	34	34	37	37	33	29	33	39	37	31
2	21	9	0	17	16	6	40	13	15	17	11	10	25	24	27	27	29	27	30	32	25	21	25	31	28	23
3	29	12	17	0	4	20	53	4	29	34	27	27	40	40	43	43	45	44	47	48	37	35	38	48	42	40
4	26	12	16	4	0	18	50	4	27	32	25	25	40	39	42	42	44	43	46	47	34	32	35	46	38	38
5	15	14	6	20	18	0	35	17	9	15	7	8	26	22	28	27	30	26	29	31	19	15	19	28	23	20
6	25	49	40	53	50	35	0	50	25	30	31	33	44	34	43	38	43	33	33	38	16	20	16	24	12	25
7	27	8	13	4	4	17	50	0	26	30	23	23	36	36	39	39	41	40	43	44	34	31	35	44	39	36
8	12	23	15	29	27	9	25	26	0	12	6	9	27	19	28	24	29	22	24	28	11	6	10	21	13	13
9	23	25	17	34	32	15	30	30	12	0	8	7	15	8	16	13	17	11	14	16	20	14	18	15	20	7
10	17	19	11	27	25	7	31	23	6	8	0	3	21	16	23	21	24	19	22	24	17	11	16	21	19	13
11	20	18	10	27	25	8	33	23	9	7	3	0	19	14	21	19	22	18	21	23	20	14	18	21	21	13
12	38	29	25	40	40	26	44	36	27	15	21	19	0	10	3	7	5	12	14	11	35	29	33	22	35	19
13	31	30	24	40	39	22	34	36	19	8	16	14	10	0	10	5	10	4	7	9	26	21	24	12	25	9
14	39	32	27	43	42	28	43	39	28	16	23	21	3	10	0	5	2	11	12	9	36	30	34	21	35	19
15	36	32	27	43	42	27	38	39	24	13	21	19	7	5	5	0	5	5	7	5	32	26	29	15	30	14
16	40	34	29	45	44	30	43	41	29	17	24	22	5	10	2	5	0	10	11	7	36	31	34	20	35	19
17	33	34	27	44	43	26	33	40	22	11	19	18	12	4	11	5	10	0	3	6	28	22	25	10	26	10
18	35	37	30	47	46	29	33	43	24	14	22	21	14	7	12	7	11	3	0	5	29	24	26	9	27	11
19	39	37	32	48	47	31	38	44	28	16	24	23	11	9	9	5	7	6	5	0	34	28	31	14	32	16
20	10	33	25	37	34	19	16	34	11	20	17	20	35	26	36	32	36	28	29	34	0	6	3	23	5	18
21	12	29	21	35	32	15	20	31	6	14	11	14	29	21	30	26	31	22	24	28	6	0	4	19	8	13
22	12	33	25	38	35	19	16	35	10	18	16	18	33	24	34	29	34	25	26	31	3	4	0	20	4	15
23	31	39	31	48	46	28	24	44	21	15	21	21	22	12	21	15	20	10	9	14	23	19	20	0	19	8
24	15	37	28	42	38	23	12	39	13	20	19	21	35	25	35	30	35	26	27	32	5	8	4	19	0	16
25	24	31	23	40	38	20	25	36	13	7	13	13	19	9	19	14	19	10	11	16	18	13	15	8	16	0

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.

3.2.3. Evaluation and update

In the SA, the evaluation is probabilistic. An SA compares the cost of the best solution and the cost of the neighboring solution. If the cost of the neighboring solution is better (less than the cost of the best solution), or if the value of $(Exp((C(x) - C(y))/T)$ is greater than the *random* (0, 1) value, the neighboring solution is assigned to the best solution ($C(x)$ is the cost of pre-best solution, $C(y)$ is the cost of neighboring solution, and T is the temperature). The cooling schedule is the decreasing function for reducing the temperature. It affects the performance of the solution and accelerates the convergence of the solution. In general, the performance time of the geometric cooling schedule is shorter than that of the logarithmic cooling schedule (Dréo et al., 2006). In this paper, we use the geometric cooling schedule to reduce the temperature quickly. The cooling ratio is set as 0.9 from the pilot experiments (i.e., $T_k = 0.9\ T_{k-1}$).

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 ($T_{initial}$), a final temperature of 1 (T_{last}), number of iterations (L) 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_{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

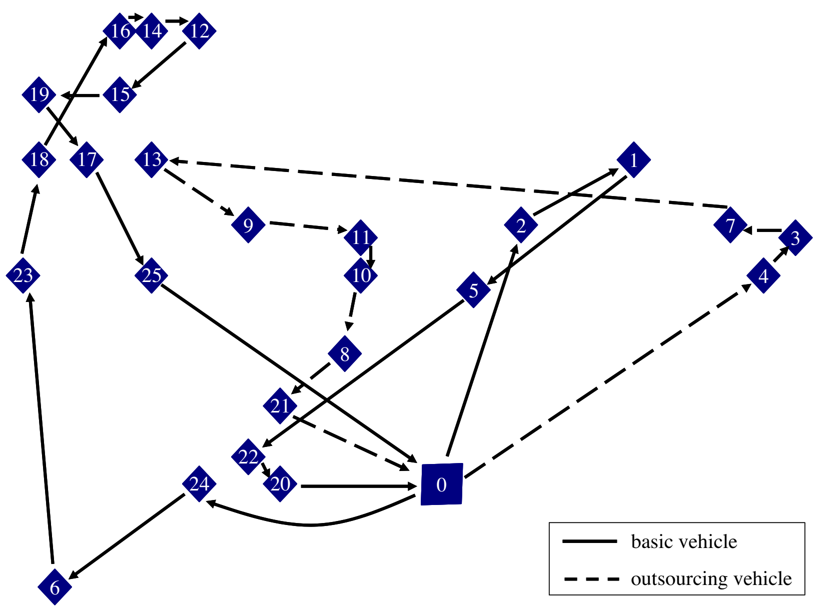

Fig. 3. Graphical results of the vehicle route (number of customers = 25).

Table 9
Comparison results of GA and hybrid algorithm (number of customers = 50).

Problem instance	Genetic algorithm				Hybrid algorithm				Remark (outperform)
	Average[a]	Min	Max	Var.	Average[a]	Min	Max	Var.	
RC201	5144.25	4821.26	5372.60	168.95	5117.67	4969.58	5272.84	105.67	Hybrid
RC202	4624.95	4464.03	4846.89	130.85	4570.38	4193.66	4878.15	230.63	Hybrid
RC203	4596.31	4251.82	4769.23	158.75	4590.41	4272.59	4813.65	167.13	Hybrid
RC204	3159.40	2486.94	3505.32	330.89	3127.28	2337.51	3484.47	307.28	Hybrid
RC205	4799.64	4681.70	4913.49	72.56	4737.27	4580.39	4901.76	118.77	Hybrid
RC206	4449.14	4267.02	4620.76	109.38	4522.93	4310.71	4880.32	150.90	GA
RC207	4105.44	3966.63	4308.10	110.34	4104.43	3963.00	4267.13	112.32	Hybrid
RC208	3430.05	3073.85	3681.49	200.02	3465.49	3275.90	3675.85	139.37	GA
C201	9747.55	9628.52	9929.39	83.82	9749.73	9593.75	9935.47	104.38	GA
C202	9439.87	9347.84	9559.80	59.94	9394.47	9226.02	9484.70	83.73	Hybrid
C203	9121.16	8482.86	9381.28	276.44	9172.13	8779.47	9343.36	191.15	GA
C204	6754.85	5961.01	7758.45	610.86	6858.78	5899.24	7894.85	711.98	GA
C205	9376.35	9155.65	9484.13	100.51	9342.35	9209.05	9462.01	87.87	Hybrid
C206	9150.67	9052.05	9207.14	50.41	9182.74	9105.48	9235.74	47.72	GA
C207	8440.32	8244.43	8500.87	82.84	8421.37	8278.58	8592.06	92.20	Hybrid
C208	8813.16	8604.29	8965.07	103.72	8829.54	8761.55	8880.23	31.30	GA

[a] Average of 10 evaluations.

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

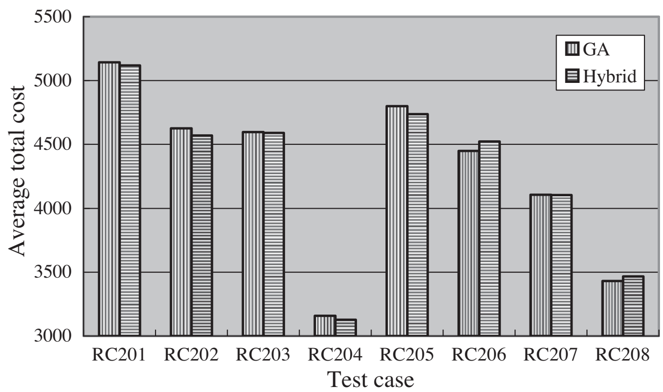

Fig. 4. Average total cost for the RC2 type instances.

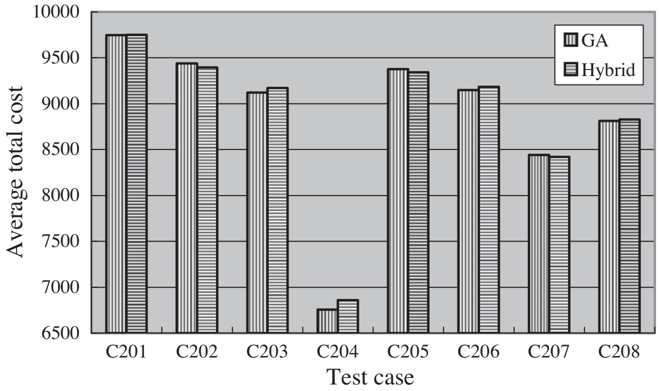

Fig. 5. Average total cost for the C2 type instances.

- Fixed cost for outsourcing vehicle: $150/vehicle
- Travel cost: $0.1/min
- Regular labor cost: $0.1/min
- Overtime labor cost: $0.2/min

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:

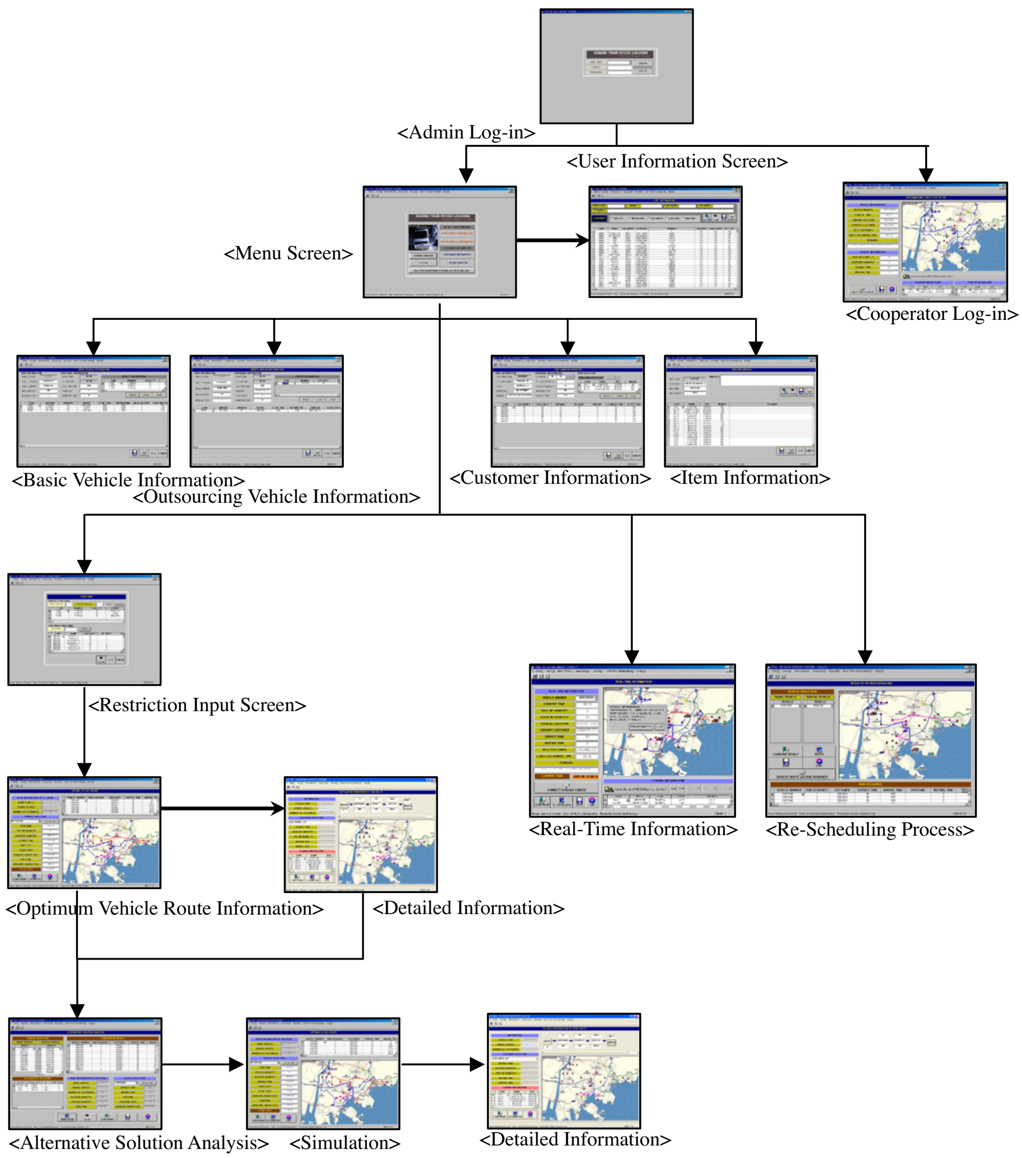

Fig. 6. Process flow of the decision support system.

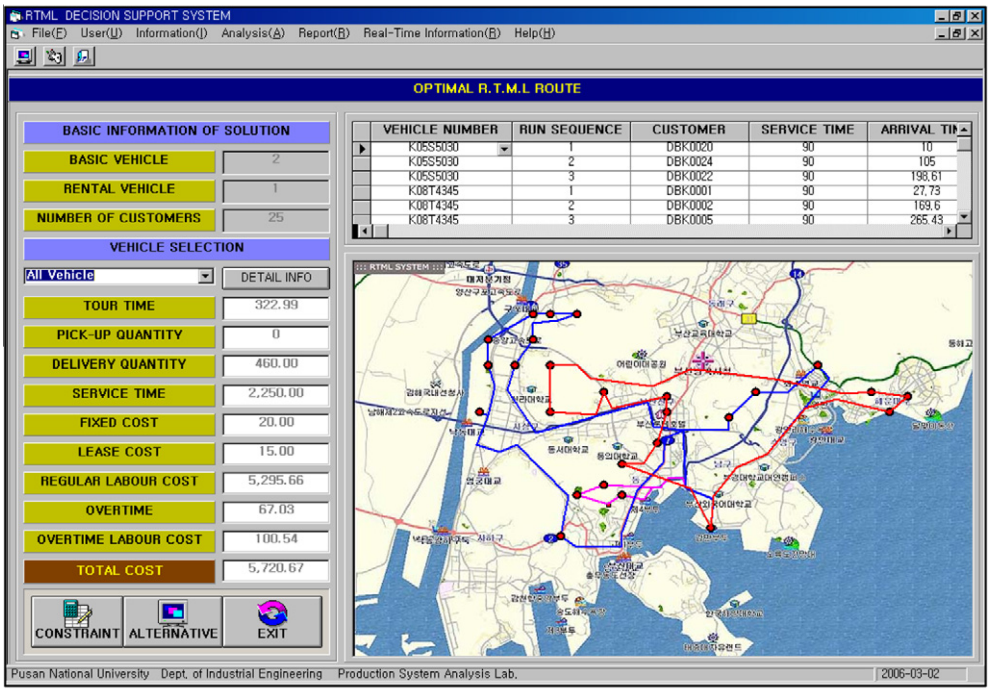

Fig. 7. Screen of optimal vehicle routes.

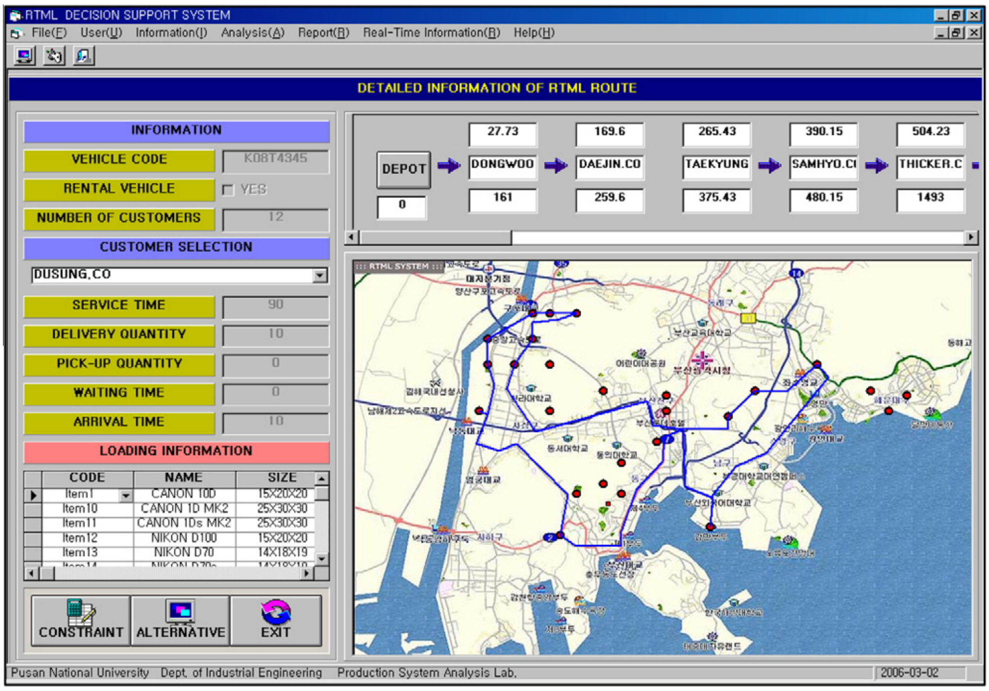

Fig. 8. Detailed information screen for a vehicle route.

- 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

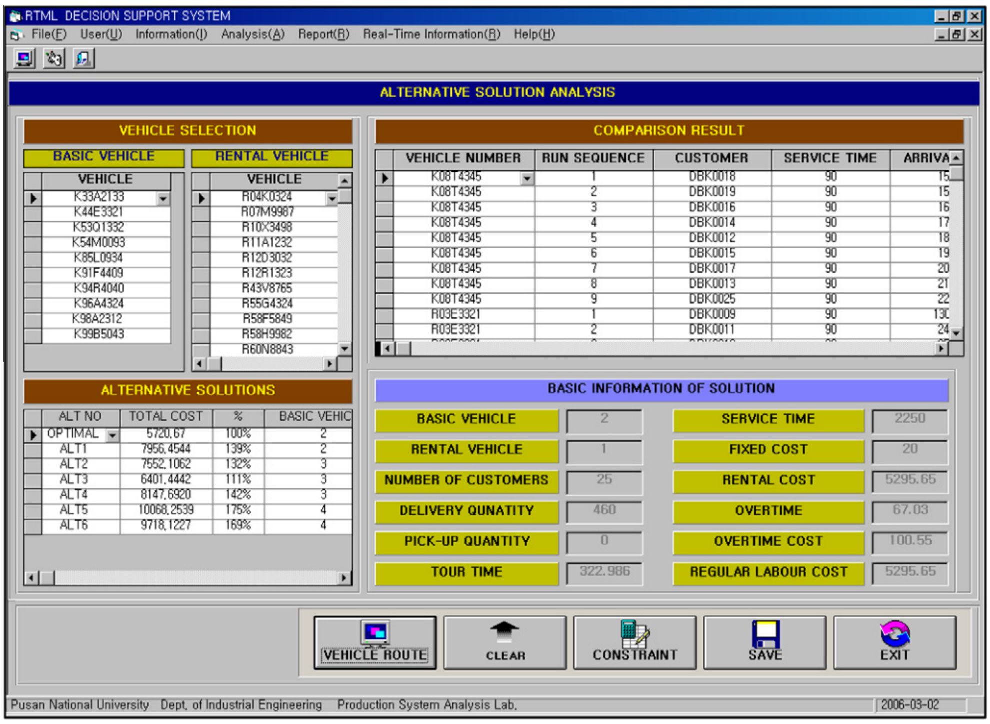

Fig. 9. Alternative solution analysis screen.

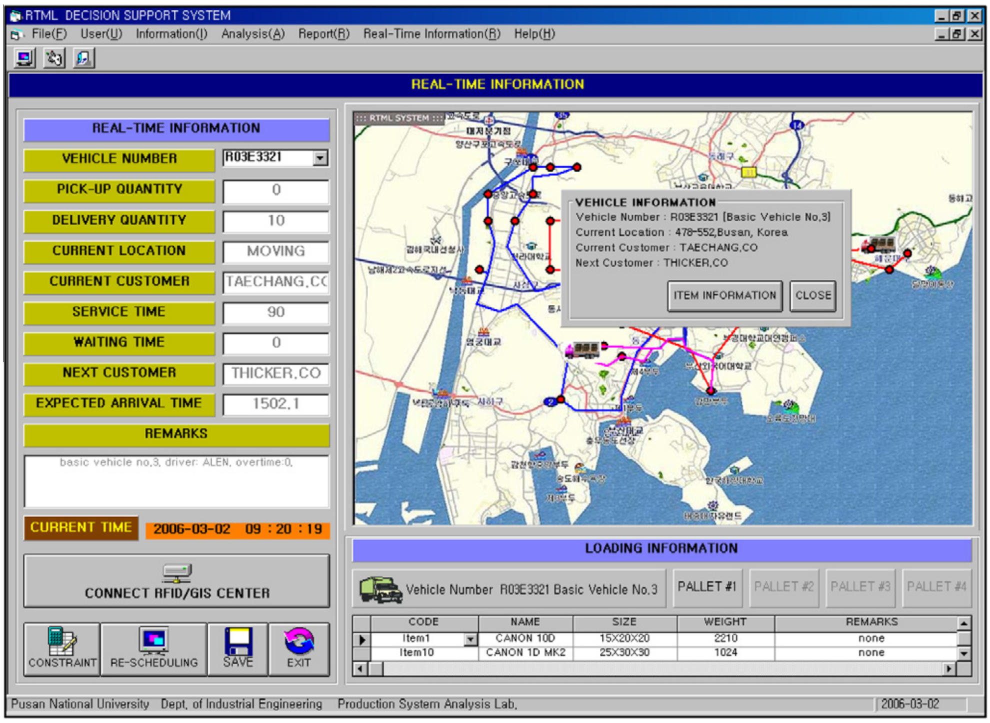

Fig. 10. Real-time information screen.

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

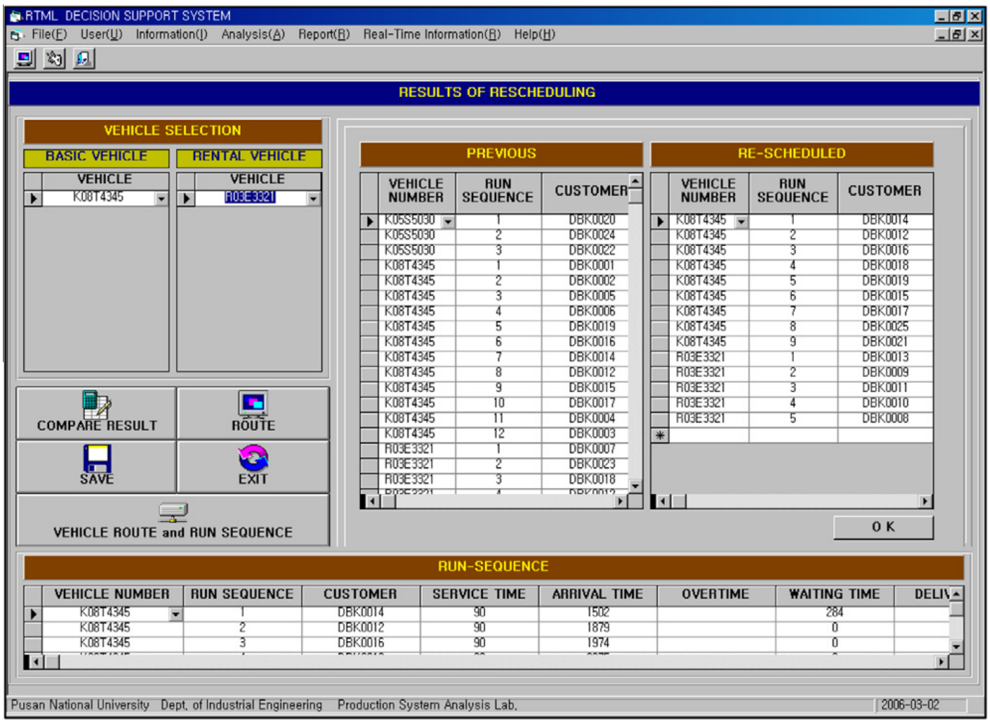

Fig. 11. Vehicle route rescheduling screen (1/2).

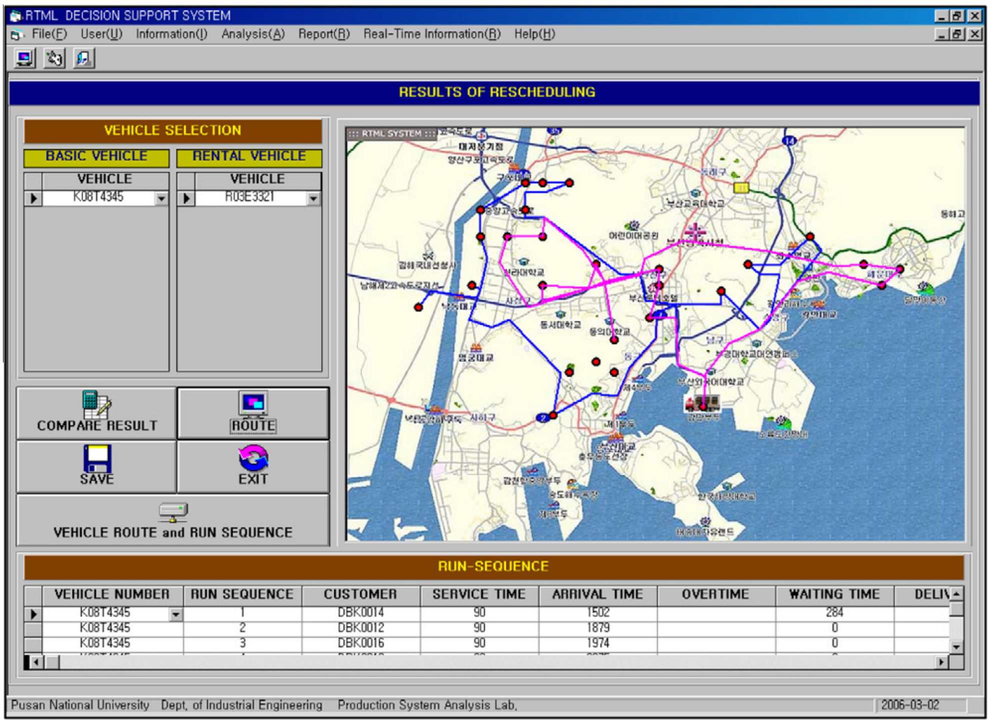

Fig. 12. Vehicle route rescheduling screen (2/2).

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 VRPTWOV. 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 (VRPTWOV) 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 VRPTWOV 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 preminary study published in the Journal of Korean Institute of Industrial Engineers in 2006.

References

Alabas-Uslu, C., & Dengiz, B. (2011). A self-adaptive local search algorithm for the classical vehicle routing problem. *Expert Systems with Applications, 38*(7), 8990–8998.

Badeau, P., Guertin, F., Gendreau, M., Potvin, J., & Taillard, E. (1997). A parallel tabu search heuristic for the vehicle routing problem with time windows. *Transportation Research Part C: Emerging Technologies, 5*, 109–122.

Bouthillier, A. L., & Crainic, T. G. (2005). A cooperative parallel meta-heuristic for the vehicle routing problem with time windows. *Computers & Operations Research, 32*, 1685–1708.

Bräysy, O., & Gendreau, M. (2005a). Vehicle routing problem with time windows – Part I: Route Construction and Local Search Algorithms. *Transportation Science, 39*, 104–118.

Bräysy, O., & Gendreau, M. (2005b). Vehicle routing problem with time windows – Part II: metaheuristics. *Transportation Science, 39*, 119–139.

Bräysy, O., Porkka, P., Dullaert, W., Repoussis, P., & Tarantilis, C. (2009). A well-scalable metaheuristic for the fleet size and mix vehicle routing problem with time windows. *Expert Systems with Applications, 36*(4), 8460–8475.

Chabrier, A. (2006). Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research, 33*, 2972–2990.

Chung, E. Y., & Park, Y. B. (2004). A genetic algorithm for vehicle routing problems with mixed delivery and pick-up. *Journal of the Korean Institute of Industrial Engineers, 30*, 346–354.

Custódio, A. L., & Oliveira, R. C. (2006). Redesigning distribution operations: a case study on integrating inventory management and vehicle routes design. *International Journal of Logistics: Research and Applications, 9*(2), 169–187.

Desrochers, M., Jones, C. V., Lenstra, J. K., Savelsbergh, M. W. P., & Stougie, L. (1999). Towards a model and algorithm management system for vehicle routing and scheduling problems. *Decision Support Systems, 25*, 109–133.

Dréo, J., Pétrowski, A., Siarry, P., & Taillard, E. (2006). *Metaheuristics for Hard Optimization*. Germany: Springer-Verlag.

Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Canada: Addison-Wesley Pub. Co., Inc.

Homberger, J., & Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *European Journal of Operational Research, 162*, 220–238.

Hwang, S. F., & He, R. S. (2006). Improving real-parameter genetic algorithm with simulated annealing for engineering problems. *Advances in Engineering Software, 37*, 406–418.

Jepsen, M., Petersen, B., Spoorendonk, S., & Pisinger, D. (2008). Subset-row inequalities applied to the vehicle-routing problem with time windows. *Operations Research, 56*, 497–511.

Kallehauge, B., Larsen, J., & Madsen, O. B. G. (2006). Lagrangian duality applied to the vehicle routing problem with time windows. *Computers & Operations Research, 33*, 1464–1487.

Karaoglan, I., & Altiparmak, F. (2011). A hybrid genetic algorithm for the location-routing problem with simultaneous pickup and delivery. *Industrial Engineering & Management Systems, 10*(1), 24–33.

Kirkpatrick, S., Gelatt, C. D., & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science, 220*, 671–680.

Lee, S. H., & Lee, J. M. (2005). Study on the heterogeneous fleet vehicle routing problem with customer restriction. *Journal of the Korean Institute of Industrial Engineers, 31*, 228–239.

Lee, L. H., Tan, K. C., Ou, K., & Chew, Y. H. (2003). Vehicle capacity planning system: a case study on vehicle routing problem with time windows. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, 33*, 169–178.

Lenstra, J., & Kan, R. (1981). Complexity of vehicle routing and scheduling problems. *Networks, 11*, 221–227.

Liu, F., & Shen, S. (1999). An overview of a heuristic for vehicle routing problem with time windows. *Computers & Industrial Engineering, 37*, 331–334.

Mester, D., & Bräysy, O. (2007). Active-guided evolution strategies for large-scale capacitated vehicle routing problems. *Computers & Operations Research, 34*, 2964–2975.

Mitchell, M. (1998). *Introduction to Genetic Algorithms*. England: MIT Press.

Nguyen, V. D., & Kim, K. H. (2010). Dispatching vehicles considering multi-lifts of Quay Cranes. *Industrial Engineering & Management Systems, 9*(2), 178–194.

Repoussis, P. P., Paraskevopoulos, D. C., Zobolas, G., Tarantilis, C. D., & Ioannou, G. (2009). A web-based decision support system for waste lube oils collection and recycling. *European Journal of Operational Research, 195*, 676–700.

Seong, J., & Moon, I. K. (2006). Vehicle routing problem with time windows considering outsourcing vehicles. *Journal of the Korean Institute of Industrial Engineers, 32*(2), 91–97.

Solomon, M. (1987). Algorithms for the vehicle routing and scheduling problem with time window constraints. *Operations Research, 35*, 254–265.

Solomon, M., & Desrosiers, J. (1988). Time window constrained routing and scheduling problems. *Transportation Science, 22*, 1–13.

Tan, K. C., Lee, L. H., Zhu, Q. L., & Ou, K. (2001). Heuristic methods for vehicle routing problem with time windows. *Artificial Intelligence in Engineering, 15*, 281–295.

Ting, C. J., & Huang, C. H. (2005). An improved genetic algorithm for vehicle routing problem with time windows. *International Journal of Industrial Engineering: Theory Applications and Practice, 12*, 218–228.

VRPTW Benchmark Problems [Online]. Available: http://web.cba.neu.edu/~msolomon /problems.htm.

Zäpfel, G., & Bögl, M. (2008). Multi-period vehicle routing and crew scheduling with outsourcing options. *International Journal of Production Economics, 113*, 980–996.

Zhang, B., & Peng, J. (2012). Uncertain programming model for chinese postman problem with uncertain weights. *Industrial Engineering & Management Systems, 11*(1), 18–25.