

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

5,500

Open access books available

137,000

International authors and editors

170M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Improvements in Simulated Quenching Method for Vehicle Routing Problem with Time Windows by Using Search History and Devising Means for Reducing the Number of Vehicles

Hisafumi Kokubugata, Yuji Shimazaki, Shuichi Matsumoto,
Hironao Kawashima and Tatsuru Daimon

Additional information is available at the end of the chapter

<http://dx.doi.org/10.5772/47854>

1. Introduction

In many countries, rationalization of freight transportation is recognized to be an important problem. For example in Japan surrounded by sea, about 60% of freight transportation is carried out by road traffic. The proportion of freight road transportation to total road transportation is close to half. Although development of information technology accelerates electronic communication, physical distribution of goods is left behind. On the contrary, because electronic commerce has enhanced door-to-door delivery services, delivery distribution of goods has increased in urban areas. The demands for high-quality delivery services such as small-amount high frequency deliveries with time windows have been made by many clients (including companies and individuals).

From the aspect of freight carrier, decrease of fuel consumption makes big profit, since the proportion of fuel to total cost is large. The rationalization in terms of increasing the loading rate and decreasing the total travel time is aimed not only for reducing operational costs in each freight carrier but also for relieving traffic congestion, saving energy and reducing exhaust gas. Effective distribution of goods should be realized by sophisticated delivery planning.

A typical delivery problem is modelled mathematically in Vehicle Routing Problem (VRP). In VRP, scattered clients are serviced only once by exactly one of plural vehicles with load

capacity which depart from a depot and return to it after touring the assigned clients. As mentioned above, clients often impose the earliest delivery time and the latest delivery time. A variation of VRP in which delivery time windows are included is called Vehicle Routing Problem with Time Windows (VRPTW). VRPTW is also applied to pick up operations such as cargo collection and garbage collection.

At the beginning of this chapter, VRP and VRPTW are introduced and followed by the explanation of precedent solution methods for VRPTW. And then, a practical solution method is proposed. It is composed by a data model, transformation rules of a solution on the data model and an overall search algorithm based on the refined Simulated Quenching (SQ) for VRPTW. The refined SQ procedures are derived from incorporating information of good solutions found in search history into basic SQ scheme. In the last section, the evaluation of the proposed method is conducted by comparisons on computational experiments with basic SQ.

2. Vehicle Routing Problem with Time Windows

Typical routing problems are abstracted from actual logistics operations in urban areas and formalized as mathematical programming problems. They are categorized as the combinatorial optimization problems.

2.1. Vehicle Routing Problem (VRP)

The Vehicle Routing Problem (VRP) is the most popular problem in routing problems. It involves the design of a set of minimum cost vehicle trips, originating and ending at a depot, for a fleet of vehicles with loading capacity that services a set of client spots with required demands. The problems studied in this chapter can be described in the style used by Crescenzi & Kann in [1] for their compendium of *NP* optimization problems. Although VRP is not listed in the compendium, it is given by Prins & Bouchenoua in [2] as follows.

- **INSTANCE:** Complete undirected graph $G = (V, E)$, initial node $s \in V$, vehicle capacity $W \in N$, length $c(e) \in N$ for each $e \in E$, demand $q(i) \in N$ for each $i \in V$, where N is the set of natural numbers.
- **SOLUTION:** A set of cycles (trips), each containing the initial node 0, that collectively traverses every node at least once. A node must be serviced by one single trip and the total demand processed by any trip cannot exceed W .
- **MEASURE:** The total cost of the trips, to be minimized. The cost of a trip is the sum of its traversed edges.

Although the VRP in a narrow sense is defined above, the VRP in a broader sense includes the more comprehensive class of routing problems related to various conditions in which demands are located on nodes. It includes VRP with time windows imposed by clients, VRP with multiple depots, periodic VRP and etc. In this case, the simplest VRP defined above is called capacitated VRP (CVRP).

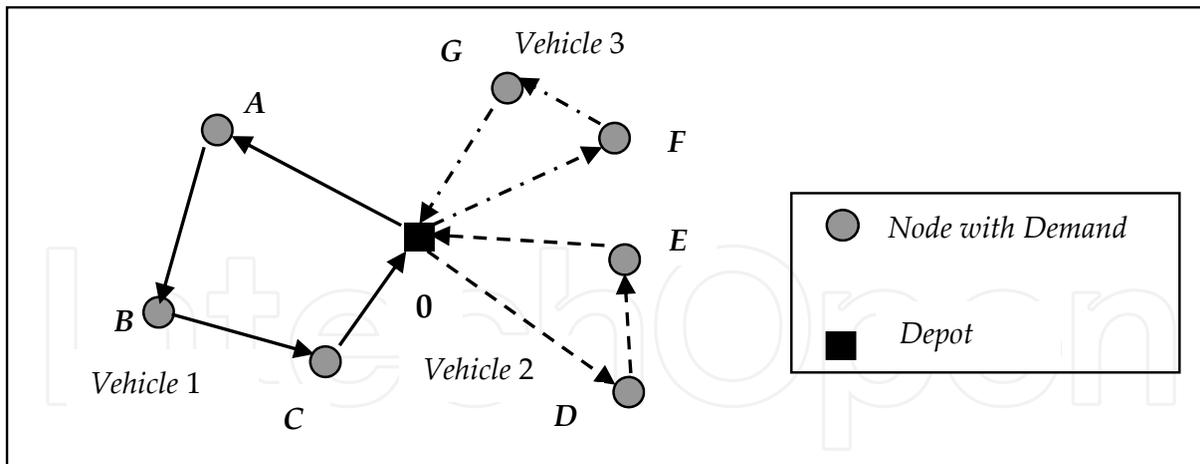


Figure 1. Vehicle Routing Problem (VRP).

2.2. Vehicle Routing Problem with Time Windows (VRPTW))

In actual delivery operations, delivery time windows are often imposed by clients. Time window at node i is described as $[e_i, l_i]$, where e_i is the earliest service starting time, l_i is the latest service starting time at node i . Vehicle routing problem taking account of time windows is called Vehicle Routing Problem with Time Windows (VRPTW). The first characteristic of VRPTW is the strictness of restriction on solutions. It often imposes increasing the number of operating vehicles. The second characteristic is the existence of waiting time. If the vehicle arrives before e_i , it must wait until e_i and then starts unloading service. Because VRP belongs to NP -hard problems, VRPTW belongs to them, too. Moreover, time windows make sequential delivery order restrictive. Hence, although both of VRP and VRPTW belong to same NP -hard problems in computational complexity theory, from a point of view with making practical algorithms, VRPTW is more difficult than VRP because of its tight constraint.

3. Precedent studies on heuristics for VRPTW

Because VRPTW belongs to NP -hard problems, exact methods are not fit for large problems. Therefore, heuristics have been important in the application of the VRPTW. Before the proposed method is explained, precedent studies on heuristics for VRPTW are introduced briefly. The heuristics for solving routing problems are classified into two major classes. The one is the family of traditional heuristics and the other is the family of metaheuristics including Simulated Annealing.

3.1. Traditional heuristics approaches for VRPTW

Comprehensive survey on traditional heuristics for VRPTW is presented in [3] by Bräysy & Gendreau. In this section, an outline of it is sketched. The traditional heuristics have been specially invented for solving VRPTW. They utilize the proper characteristics of VRPTW. They are further classified into two types.

The first one is the type of constructive heuristics that produce vehicle routes by merging existing routes or inserting nodes into existing routes. Ioannou et al. proposed an efficient constructive heuristic in [4]. They use the generic sequential insertion framework proposed in [5] by Solomon.

The second one is the type of improvement heuristics which make changes in one vehicle route or between several vehicle routes. Bräysy proposed several efficient local search heuristics in [6] using a three-phase approach. In the first phase, several initial solutions are created using the route construction heuristics with different combinations of parameter values. In the second phase, an effort is put to reduce the number of routes. In the third phase, classical Or-opt exchanges, which replace three edges in the original tour by three new ones without modifying the orientation of the route, are used to minimize total travelled distance.

3.2. Metaheuristics for VRPTW

Metaheuristics have been introduced into the solutions for VRPTW in the last two decades. Because metaheuristics are generally recognized to fit combinatorial optimizations, Simulated Annealing (SA), Tabu Search (TS), Genetic Algorithm (GA) and Ant Colony Optimization (ACO) have been tried to apply to VRPTW. Traditional heuristics explained in Sec. 3.1 are often embedded into these metaheuristics.

Cordeau et al. presented an efficient TS heuristic in [7]. Among the methods incorporating GA, the methods proposed by Homberger & Gehring in [8] and Berger et al. in [9] are reported to get good results. With respect to ACO, although not so many works on VRPTW are appeared in the literature, Gambardella et al. use an ACO approach with a hierarchy of two cooperative artificial ant colonies in [10]. Chiang & Russell developed a SA approach for VRPTW in [11]. They combined the SA process with the parallel construction approach that incorporates improvement procedures during the construction process.

In a comprehensive survey on metaheuristics for VRPTW given by Bräysy & Gendreau in [12], it is described that some hybrid methods are very effective and competitive with two good GA algorithms listed above. They are briefly introduced as follows. Bent & Van Hentenryck present a two-stage hybrid metaheuristic in [13], where in the first stage is a basic SA used to minimize the number of routes, and the second stage focuses on distance minimization using the large neighbourhood search. Bräysy presents a four-phase deterministic metaheuristic algorithm in [14] which is based on a modification of the variable neighbourhood search. Ibaraki et al. propose three metaheuristics in [15] to improve randomly generated initial solutions.

4. Data model and method of generating neighbours in searching process of simulated quenching for VRPTW

Although some precedent methods based on metaheuristics mentioned above show good performance, their procedures are considerably complex. In particular, the local search

procedures incorporated into them are rather complicated. In practical application of VRPTW algorithms to real-world problems, ease of implementation and flexibility are very important as well as quality of solution, running time and robustness. Hence, the authors of this chapter have proposed a simpler data model and a one-phase algorithm to solve VRPTW in [16] which is not the two-phase algorithm composed by construction and improvement.

4.1. Data modelling for VRPTW

The model to express a state of solution of VRPTW is realized as a sequence of integers, i.e., a string. In the string, the position of an integer, which is a symbol of the node with demand, implies not only which vehicle tours the node but also the routing order of it. An example of the string model is illustrated in Figure 2. The special number '0' should be interpreted not only as the depot but also as the delimiter which partitions the trips. If the number of vehicles is denoted by m , $(m-1)$ '0's are provided in the string. If there is no integer between '0' and '0', the relevant vehicle is not in use.

This data model is coincidentally similar to that invented for the solution based on a kind of GA. It was introduced by Gendreau et al. in [17] as the original idea was given by Van Breedam in [18]. However, the proposed transformation rules in this chapter based on the data model are quite different from those of precedent methods. They will be described in the following section.

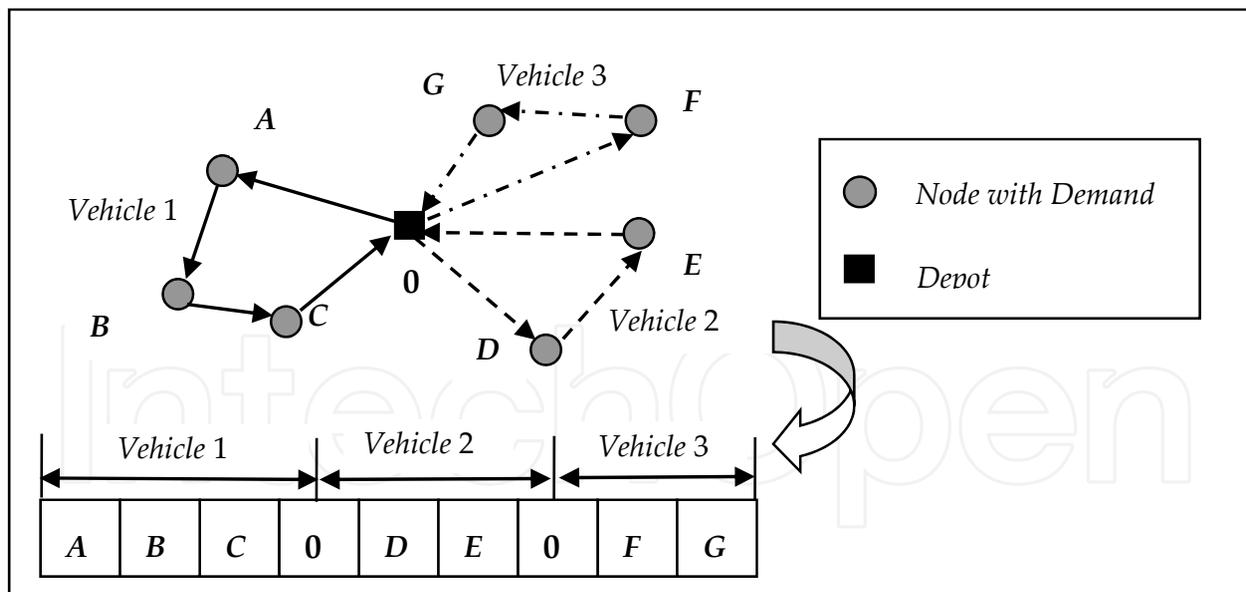


Figure 2. Proposed data model for VRPTW.

4.2. Transformation rules for generating neighbours

In a repetition in the proposed procedure, a new state of solution is generated from the present state by one of the following three types of transformation rules for generating neighbours. The first rule is to exchange an integer with another one in the string. The

second rule is to delete an arbitrary integer and then insert it to another position in the string. The third rule is that after a substring is taken out temporarily, the direction of the substring is reversed, and then embedded in the place where the substring is taken out. These three transformation rules are illustrated in Figure 3.

Note that the rules are also applied to the special number '0' in the string illustrated in Figure 2. In other words, '0' is treated impartially with other integers. If 'one-to-one exchange' is executed within a substring partitioned by two '0's, only a route is changed. An example of the case is illustrated in Figure 4. If 'one-to-one exchange' is executed between two non-zero integers striding over '0', two nodes are exchanged between two routes. An example of this case is illustrated in Figure 5. If 'one-to-one exchange' is executed between a non-zero integer and '0', two routes are merged, while another route is divided into two routes. An example is illustrated in Figure 6.

When the second transformations rule 'delete and insert' is applied, several different cases also arise. If a non-zero integer is deleted and inserted at '0', a node is moved to another vehicle route. An example is illustrated in Figure 7.

When the third transformations rule 'partial reversal' is applied, several different cases also arise. If a substring including '0' is reversed, the relevant plural routes are changed. An example is illustrated in Figure 8. These three transformation rules were originally invented for VRP in [19] by the authors of this chapter.

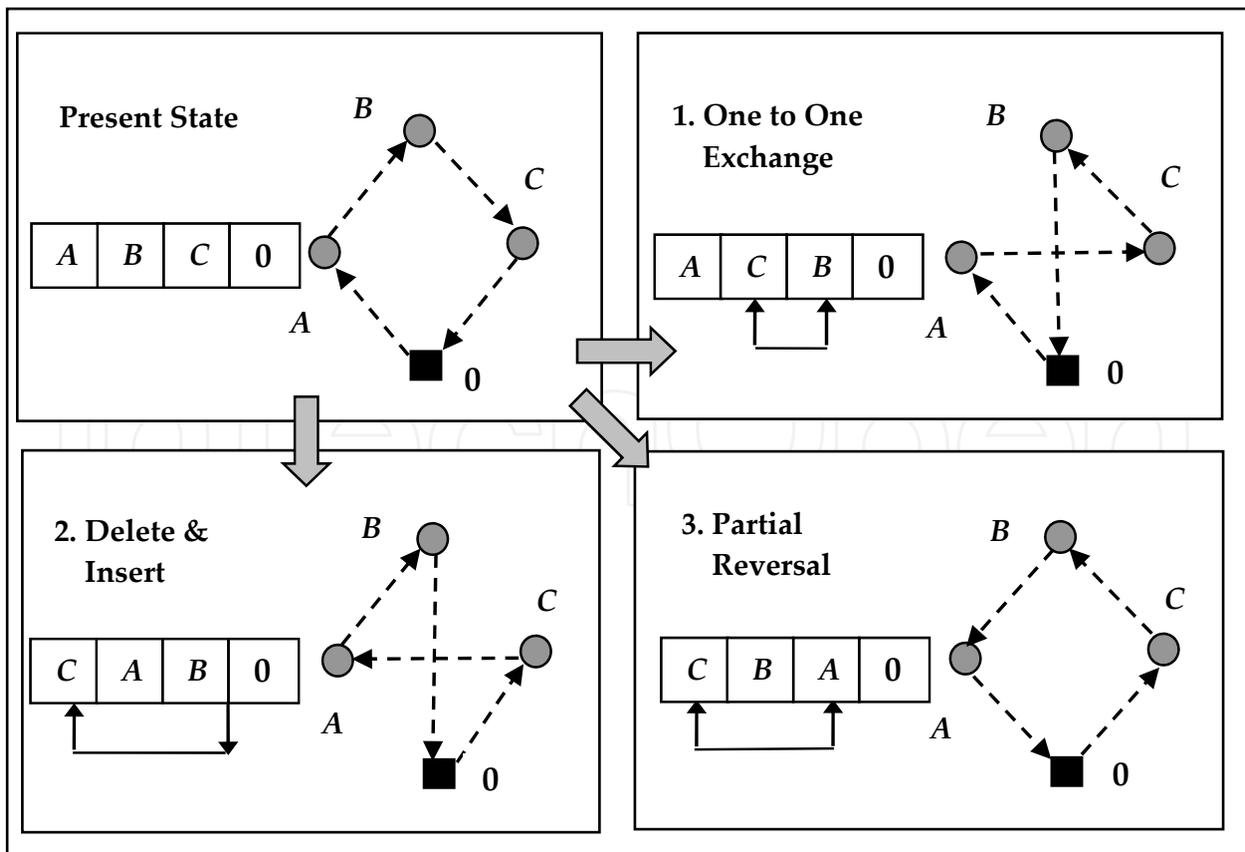


Figure 3. Transformation rules for generating neighbours.

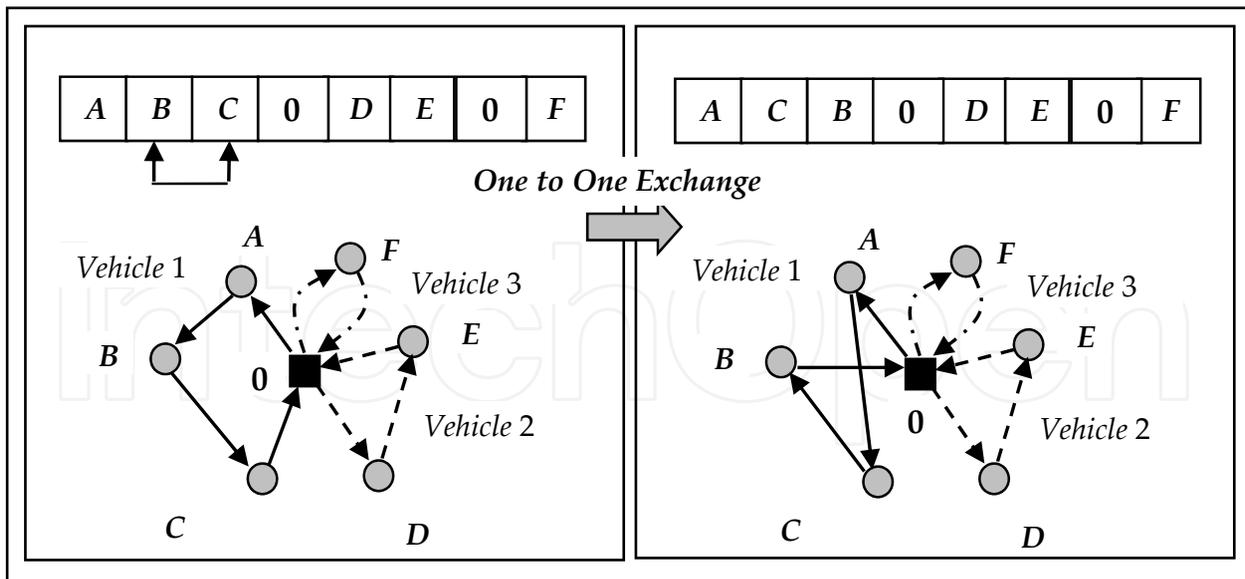


Figure 4. A result of 'one-to-one exchange' within a route.

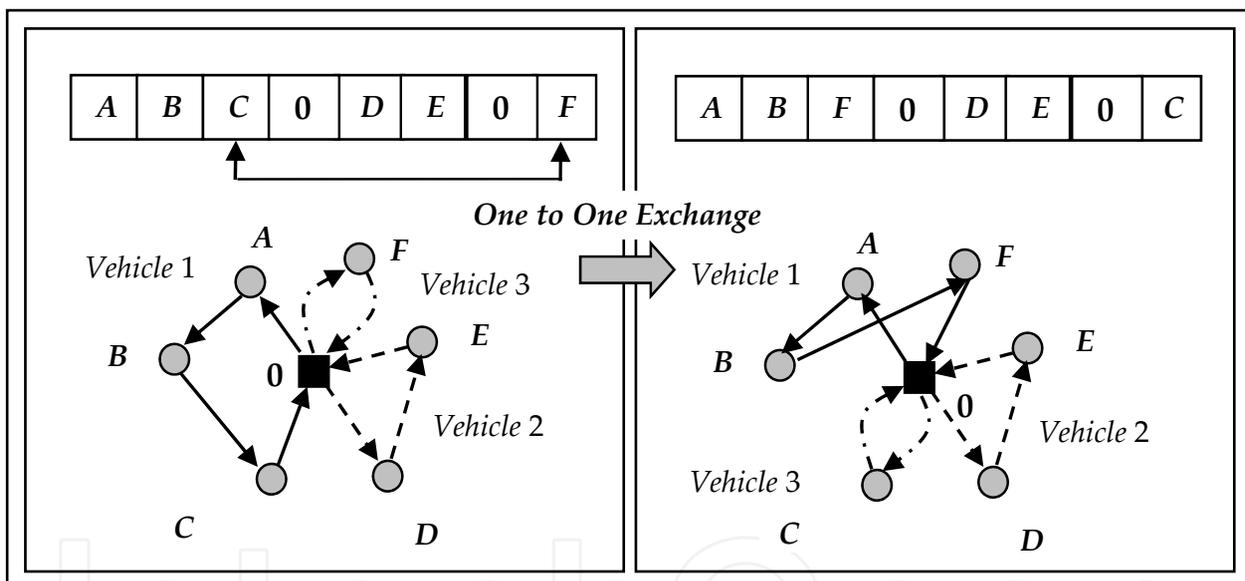


Figure 5. A result of 'one-to-one exchange' between two non-zero integers striding over '0'.

4.3. Objective function

The objective of the VRPTW is the minimization of total cost which is subject to constraints including the loading capacity of each vehicle and the time windows imposed by clients. The objective function of the VRPTW is formulated as follows.

$$E(s) = \sum_{i=1}^n c_{s_i} + \sum_{i=0}^n d_{s_i, s_{i+1}} \quad (1)$$

where $s = (s_1, s_2, \dots, s_n)$ is a string that consists of the nodes with demands and a depot; s_0 and s_{n+1} are the implicit expressions of the depot omitted in the string s ; c_k is the servicing

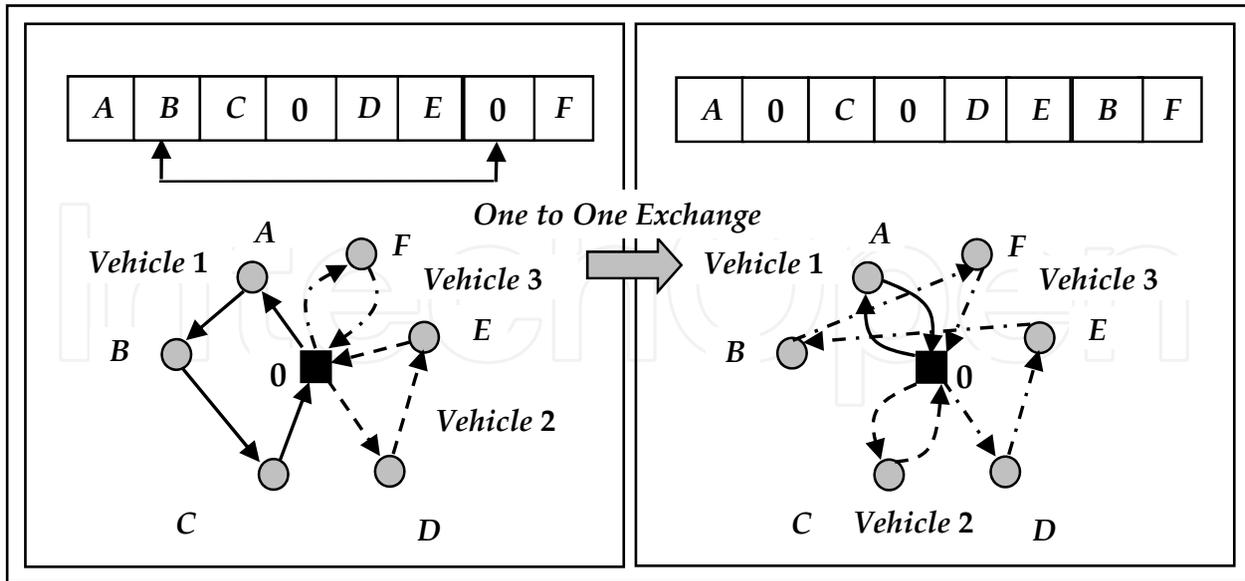


Figure 6. A result of ‘one-to-one exchange’ between non-zero integer and ‘0’.

cost at the node k (if $k = 0$, then $c_k = 0$); $d_{k,l}$ is the minimal traversing cost from the node k to the node l . Each value of $d_{k,l}$ might be given by input data; or calculated as the Euclidean distances between a pair of coordinates of nodes; or calculated by the shortest path search algorithm (Warshall-Floyd’s algorithm) when road network is given and vehicles must follow the roads in the network.

In order to impose solutions of VRPTW to satisfy time window constraints and load capacities and to reduce the number of vehicles in use, three penalty terms are added to the objective function (1) as follows:

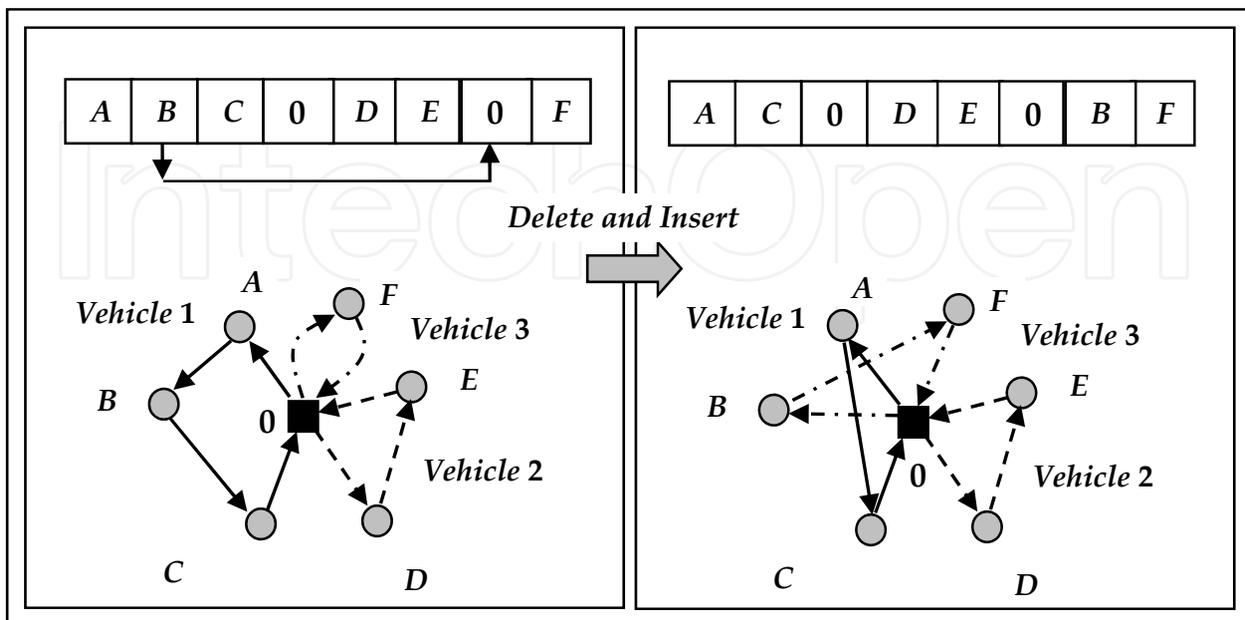


Figure 7. A result of deleting non-zero and inserting it at ‘0’.

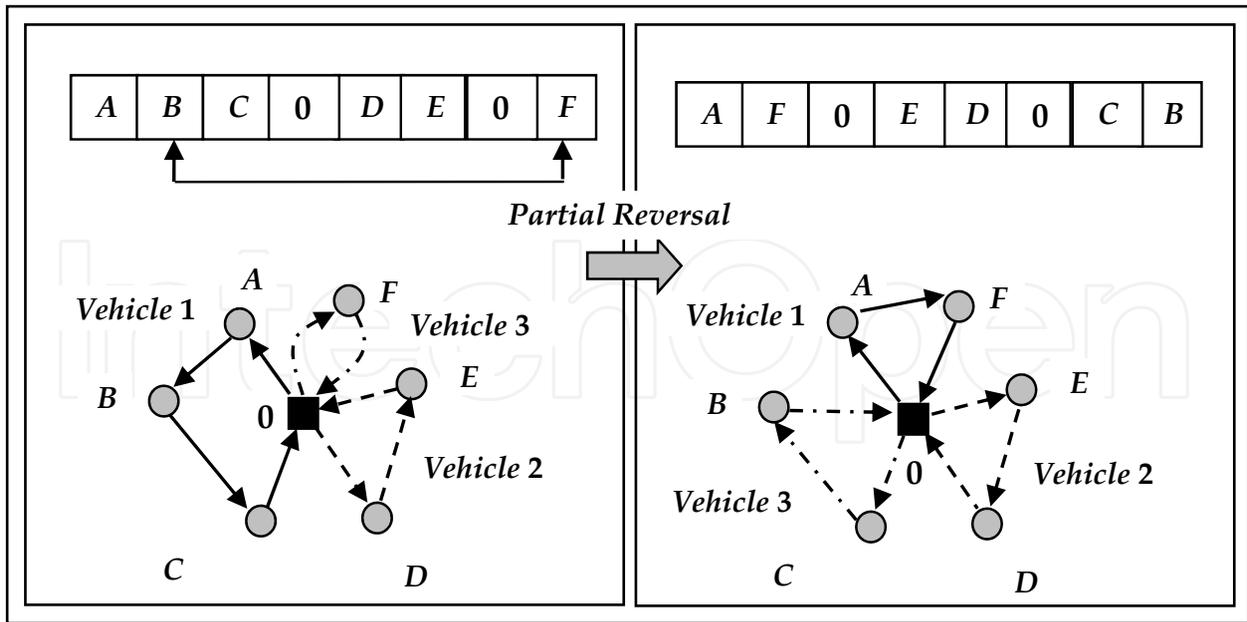


Figure 8. A result of ‘partial reversal’ striding over ‘0’.

$$E'(s) = \left(\sum_{i=1}^n c_{s_i} + \sum_{i=0}^n d_{s_i, s_{i+1}} \right) + \alpha \left(\sum_{i=1}^{n+1} \max(0, a_{s_i} - l_{s_i}) \right) + \beta \left(\sum_{k=1}^m \left(\sum_{i=z_{k-1}+1}^{z_k} w_{s_i} - W_k \right) \right) + \gamma m \quad (2)$$

where a_{s_i} is arriving time at node s_i ; l_{s_i} is the latest service starting time at node s_i ; m is the number of vehicles in use; w_{s_i} is the amount of demand at node s_i ; z_k is the position of k th ‘0’ in the string $s = (s_1, s_2, \dots, s_n)$ (provided that $z_0 = 0$; $z_m = n+1$) and W_k is the loading capacity of vehicle k . α , β and γ are weight parameters.

4.4. Optimization algorithm using Simulated Quenching

Simulated Quenching is adopted as the optimization technique for the proposed method since it is characterized by simple stochastic procedures and by global searching scope. In the original Simulated Annealing (SA), starting with a random initial state, it is expected to approach an equilibrium point. In order to obtain global optimum, cooling schedule should be logarithmic. However, it spends too much time to implement it. Hence, in practical applications, exponential cooling schedule (3) is often adopted.

$$T(t+1) := kT(t) \quad (0 < k < 1) \quad (3)$$

In the proposed method, it is adopted too. According to the strict theory of Simulated Annealing, the optimization technique using exponential cooling schedule (3) belongs to Simulated Quenching (SQ) as described in [20]. SQ is considered to be a practical variant of SA.

In the proposed method, the three transformation rules described in Sec. 4.2 are applied randomly to the string model. The entire algorithm for the VRP is described as follows.

{I. Preparation}
 Read input data;
 If the link cost are not given from the input data, calculate the minimum path cost $d_{i,j}$ between all pair of clients i, j including the depot 0;
 {II. Initialization} Generate an initial feasible solution s_0 by assigning nodes to vehicles in ascending order of the specified earliest arriving time; $s := s_0$; $s^* := s$;
 $T := \text{INITTEMP}$; Set N as the averaged neighbourhood size;
 {III. Optimization by SQ} Minimize E by repetition of applying randomly one of the three transformation rules to the string model corresponding to x in the framework of SQ;
 {IV. Output} Output the best solution s^* .

(4)

Step III, that is the main part of this algorithm, is detailed as follows.

Repeat
 trials := 0; changes := 0;
 Repeat
 trials := trials + 1;
 Generate a new state s' from the current state s by applying randomly one of the three transformation rules to the string model of s ;
 Calculate $\Delta E' = E'(s') - E'(s)$;
 If $\Delta E' < 0$ Then
 s' is accepted as a new state;
 If $(E'(s') < E'(s^*)$ and s' is feasible) Then $s^* := s'$
 Else s' is accepted with probability $\exp(-\Delta E'/T)$
 If s' is accepted Then changes := changes + 1; $s := s'$
 Until trials \geq SIZEFACTOR $\cdot N$ or changes \geq CUTOFF $\cdot N$;
 $T := T \cdot \text{TEMPFACTOR}$
 Until $T \leq \text{INITTEMP} / \text{FINDIVISOR}$

(5)

The words noted by capital letters are parameters used in SA and SQ and values of them are specified in Sec 6.2. As described in Sec. 4.2, the transformation procedure of a solution of the proposed method is carried out randomly to all over the string data model. Hence, the transformation might derive changes in a vehicle route on one occasion, it might derive changes over several vehicle routes on other occasion. This method is applied to VRP in [19], VRP with backhauls (VRPB) in [21], Pick up and Delivery Problem (PDP) and VRPTW in [16] by the authors of this chapter. It is also applied to other types of routing problems including Capacitated Arc Routing Problem (CARP) in [22] and a general routing problem with nodes, edges, and arcs (NEARP) in [23]. A precise analysis of this method is presented in [24].

5. Improvement of optimization algorithm based on SQ by adaptation of devices inspired by ACO

Most of metaheuristics belong to stochastic local search (SLS) which starts at some position in search space and iteratively moves to neighbour, using randomised choices in generating and modifying candidates. In application of metaheuristics, both intensification and

diversification are important. For sufficient convergence of solution, intensification of search scope is necessary. On the other hand, in order to avoid stagnation in local but not global minimum area, diversification of search scope is also necessary. As compared to 'stupid fly', search process in pure SA and SQ is completely random. Making use of history records during search processes might be possible to improvement of optimization algorithm based on SQ.

5.1. Ant Colony Optimization

ACO was introduced by Dorigo et al. in [25]. It was inspired by foraging behaviour of ants. Ants often communicate via chemicals known as pheromones, which are deposited on the ground in the form of trails. With time, paths leading to the more interesting food sources become more frequented and are marked with larger amounts of pheromone. Pheromone trails provide the basis for stochastic trail-following behaviour underlying, for example, the collective ability to find shortest paths between a food source and the nest. ACO is described as follows.

Initialise pheromone trails;

While termination criterion is not satisfied

Generate population sp of candidate solutions (6)

using subsidiary randomised constructive search

Perform subsidiary local search on sp ;

Update pheromone trails based on sp

In applying ACO to TSP (Travelling Salesman Problem) which is single vehicle version of VRP, details are specified as follows.

1. Pheromone trail τ_{ij} is associated with each edge (i, j) in G , while heuristic values $\eta_{ij} = 1 / d_{ij}$ is used, where d_{ij} is traversing cost of edge (i, j) .
2. In the beginning, all weights are initialised to small value τ_0 .
3. In constructive search, each artificial ant starts with randomly chosen node and iteratively extends partial round trip φ by selecting node not contained in φ with probability:

$$\frac{[\tau_{ij}]^a \cdot [\eta_{ij}]^b}{\sum_{l \in N'(i)} [\tau_{il}]^a \cdot [\eta_{il}]^b}, \quad (7)$$

where $N'(i)$ is the feasible neighbourhood of node i , that is, the set of all neighbours of i that are not contained in φ . a and b are parameters which control the relative impact of the weights *vs.* the heuristic values.

4. After the constructive search, subsidiary local search which is iterative improvement based on standard 2-exchange neighbourhood is operated until local minimum is reached.

5. In the end of loop, pheromone trail is updated according to

$$\tau_{ij}(t+1) := (1 - \rho) \cdot \tau_{ij}(t) + \sum_{k=1}^m \Delta \tau_{ij}^k, \quad (8)$$

where $0 < \rho \leq 1$ is a parameter regarding vaporization of pheromone,

$$\Delta \tau_{ij}^k = \begin{cases} Q/E(s^k) & \text{if } (i, j) \in s^k \\ 0 & \text{else} \end{cases}. \quad (9)$$

$E(s^k)$ is total cost of the k th ant's cycle s^k , m is the number of ants (= size of sp) and Q is a constant. Criterion for weight increase is based on intuition that edges contained in short round trips should be preferably used in subsequent constructions.

As mentioned in Sec. 3.2, not so many methods based on ACO for VRPTW are appeared in the literature.

5.2. Application of information obtained in search history to SQ

One of main drawbacks of SA and SQ which are pointed out by users of other metaheuristics is lack of learning in search history, that is, blind random search often compared to 'stupid fly'. Although the complete ACO belongs population-based SLS methods in which genetic algorithm is also contained, a predecessor of ACO is Ant System which is a single ant version of ACO. It was also proposed by Dorigo et al. in [25] and it is recognized as a member of Adaptive Iterated Construction Search (AICS) methods. In the Ant System single artificial ant works and uses information obtained in its own preceding searches. Utilization of some information about good solutions obtained in the preceding search processes is able to be incorporated into SQ procedures. It would be possible to overcome the blind random searches in SQ. Because traversed arcs in good solutions of VRPTW are recognized as characteristics of them, such arcs are expected to be not drastically changed in the succeeding search processes.

In order to embody the idea described above, artificial pheromone trail τ_{ij} is associated with each edge (i, j) and τ_{ij} is updated at the end of the loop of temperature T in SQ procedures. The 'characteristic of good solution' is embodied in increase of probability of selecting better candidate in random search process in SQ. In end of loop, weight is updated according to

$$\tau_{ij}(t+1) := (1 - \rho) \cdot \tau_{ij}(t) + \Delta \tau_{ij} \quad (10)$$

, where $0 < \rho \leq 1$ is a parameter regarding vaporization of pheromone,

$$\Delta \tau_{ij} = \begin{cases} Q/E(s^*) & \text{if } (i, j) \in s^* \\ 0 & \text{else} \end{cases}. \quad (11)$$

However, in order to avoid extreme effect of pheromone, lower bound and upper bound of τ_{ij} is set as $0.2 \leq \tau_{ij} \leq 1.4$. $E(s^*)$ is the total cost of the best found solution at the present s^* , Q is a

constant. Let $r_i = d_{ai} / \tau_{ai} + d_{ib} / \tau_{ib}$, where d_{ij} is traversing cost of edge (i, j) . When edges (a, i) and (i, b) which are connected with node i are frequently contained in the best routes and costs of the edges are small, the value of r_i becomes small. Because such a situation of node i is agreeable to good solutions, it should not be drastically changed in the succeeding search processes. In order to embody the idea stated before, r_i is used for assigning node i the biased small probability with which node i is selected for change, instead of obeying uniform distribution as described in Sec. 4.2. That is to say, node i is selected for transformation with probability:

$$p_i = \frac{r_i}{\sum_i r_i} \quad (12)$$

The core part of the basic SQ algorithm (5) is replaced by the revised algorithm which is called SQ_{ph} described as follows.

$\tau_{ij} := 0.2$ for all (i, j) ;
 $p_i := 1 / \sum_i r_i$ for all i ;
Repeat
 trials := 0; *changes* := 0;
 Repeat
 trials := *trials* + 1;
 Generate a new state s' from the current state s by applying randomly one of the three transformation rules to the string model of s , where position i to be changed in the string is selected with the probability p_i ;
 Calculate $\Delta E = E(s') - E(s)$;
 If $\Delta E < 0$ Then
 s' is accepted as a new state;
 If $(E(s') < E(s^*))$ and s' is feasible) Then $s^* := s'$
 Else s' is accepted with probability $\exp(-\Delta E/T)$
 If s' is accepted Then *changes* := *changes* + 1; $s := s'$
 Until *trials* \geq SIZEFACTOR $\cdot N$ or *changes* \geq CUTOFF $\cdot N$;
 T := *T* \cdot TEMPFACTOR;
 $\tau_{ij} := (1 - \rho) \cdot \tau_{ij} + \Delta \tau_{ij}$; if $\tau_{ij} > 1.4$ then $\tau_{ij} := 1.4$; elseif $\tau_{ij} < 0.2$ then $\tau_{ij} := 0.2$ for all (i, j) ;
 $r_i := d_{ai} / \tau_{ai} + d_{ib} / \tau_{ib}$, $p_i := r_i / \sum_i r_i$ for all i ;
 Until $T \leq$ INITTEMP / FINDIVISOR

5.3. A device for decreasing the number of vehicles in use

When performance of plural solutions for VRPTW is compared, the first measure is the number of vehicles in use, which is denoted by m , while the second is total cost E . Although SQ_{ph} is expected to utilize characteristics of good solutions already found and to reduce E , it could not reduce m directly. In order to reduce it, another device should be included in SQ procedures.

In the string model described in Sec. 4.1, successive substring of '0's is interpreted as there is no tours between two '0's, that is to say that there is a vehicle not in use. For example shown in Figure 9, when '0' is replaced by another symbol in the string, one vehicle will become not in use. In order to urge to carry out this kind of transformation, artificial pheromone trail associated with edges $(i, 0)$ or $(0, i)$ should be decreased.

$$\tau_{i0}(t+1) := \{(1 - \rho)\tau_{i0}(t) + \Delta\tau_{i0}(t)\} \times \delta, \quad \tau_{0i}(t+1) := \{(1 - \rho)\tau_{0i}(t) + \Delta\tau_{0i}(t)\} \times \delta, \quad 0 \leq \delta \leq 1 \tag{14}$$

The effect of the device (14) could make the probability p_0 large according to mechanism described in Sec.5.2. This further revised algorithm in which the device (14) is incorporated with pheromone update (10) is called SQ_{ph}^* in this chapter.

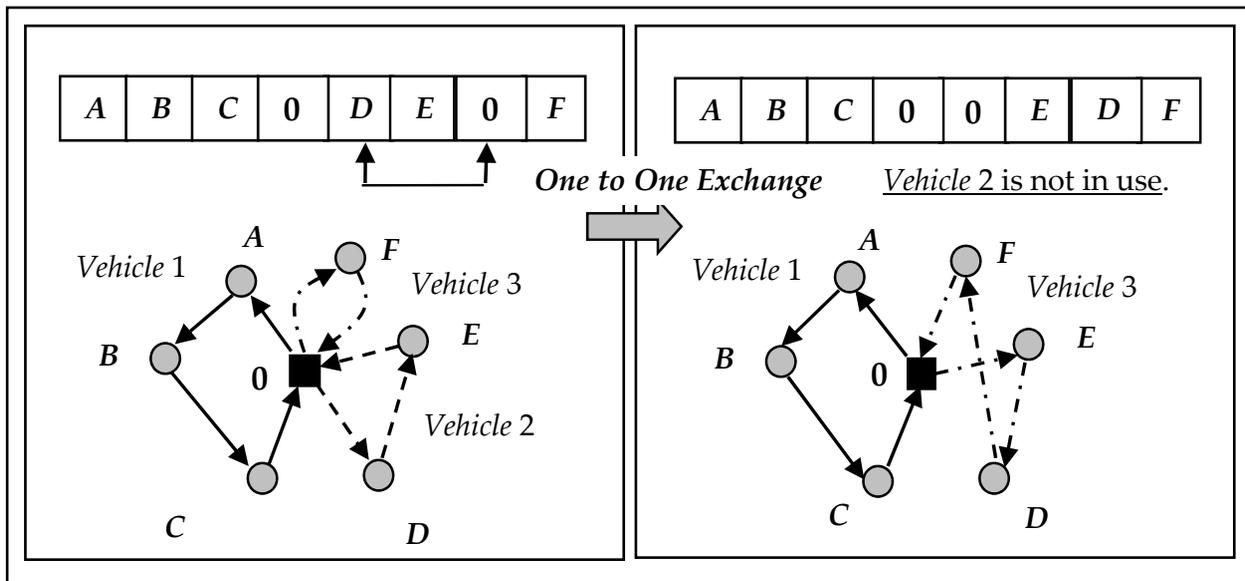


Figure 9. Reduction of m as a result of one to one exchange including '0'.

6. Computational experiments on the proposed method

Computational experiments have been attempted for testing the performance of the proposed method compared with basic SQ method. They have been tried on typical instances for VRPTW.

6.1. Solomon's benchmark problems and extended problems for VRPTW

Solomon's benchmark problem sets are produced by Solomon in [5] and provided from Solomon's own website in [26]. They are extremely popular VRPTW instances, and have been used for testing performance of methods by many researchers. Although in some of instances, optimum solutions have been already found by using exact methods, in others, they have not found yet. In both cases, the best solutions found by heuristics have been presented in the literature. Instances including 25, 50, and 100 clients have been provided

from Solomon's problem sets. In the instances, each position of clients is given as x -coordinate and y -coordinate. Link cost between client i and client j is calculated with the Euclidian distance. Service time c_i is also given to each client i , in addition to the earliest arriving time e_i and the latest arriving time l_i . In this benchmark problems, link cost is directly considered as traversing time of (i, j) . Arriving time a_i of each node i is calculated by summing up link cost of traversing edges and service time of traversing nodes. Concerning 100 clients problem sets, the geographical data are clustered in C-series 23 instances, randomly generated in R-series 17 instances, and a mix of random and clustered structures in RC-series 16 instances.

Gehring & Homberger extended Solomon's benchmark problems to larger scale problem sets including 200, 400, 600, 800 and 1000 clients in [27]. They are provided from their website [28]. Concerning 200 clients problem sets, the geographical data are clustered in C-series 20 instances, randomly generated in R-series 19 instances, and a mix of random and clustered structures in RC-series 20 instances.

In this chapter, all of 56 instances with 100 clients from Solomon's problems and all of 59 instances with 200 clients from Gehring & Homberger's problems are chosen for computational experiments.

6.2. Values of parameters used in the algorithms and specs of the computer in use

In the computations, the values of the parameters with respect to SQ that appear in the basic SQ, SQ_{ph} and SQ_{ph}^* are set commonly as follows according to the preliminary experiments and the reference to the recommended values by Johnson et al. in [29-30].

$N = 2L^2$ (L : length of string, that is $L = n + vn - 1$, where n is the number of clients, vn is the number of vehicles superfluously allocated)

$SIZEFACTOR = 8$

$CUTOFF = 0.2$ (Repeat iterations in the same temperature T ,
until ($trials \geq SIZEFACTOR \cdot N$ or $changes \geq CUTOFF \cdot N$))

$INITTEMP = 10$ (Initial temperature)

$$TEMPFACTOR = 0.95 \quad (T_{n+1} = 0.95T_n) \quad (15)$$

$FINDIVISOR = 20$ (If $T \leq INITTEMP / FINDIVISOR$, terminate the whole of the iterations.)

Values of parameters appeared in energy function (2)

$$\alpha = 25, \beta = 1, \gamma = 500 ; \quad (16)$$

and values of parameters used in the proposed method

$$\rho = 0.5 \text{ in (10) and (14); } Q = 1000 \text{ in (11)} \quad (17)$$

are set according to the preliminary execution.

The computational experiments are executed on Windows 7, with Core i7 960, 3.2GHzCPU.

6.3. Comparison between basic SQ and SQ_{ph}

Ratio of application of artificial pheromone trail τ_{ij} to all transformations is set to 3 cases regarding 50%, 75% and 100%. In other words, the ratio of random transformation is 50%, 25% and 0% in each case. Computational experiments are performed ten times on all benchmark problems with 100 clients and 200 clients.

Concerning the number of vehicles in use m , significant difference is not detected. Regarding total traversing cost E , improvement ratio of SQ_{ph} to SQ is illustrated in Figure 10. Values corresponding to the best cost solutions and the worst cost solutions of SQ in ten executions are shown by two bars. Computing time consumed by the methods using SQ and SQ_{ph} is illustrated in Figure 11. According to these results, larger ratios of application of artificial pheromone trail bring better total traversing cost and longer computing time.

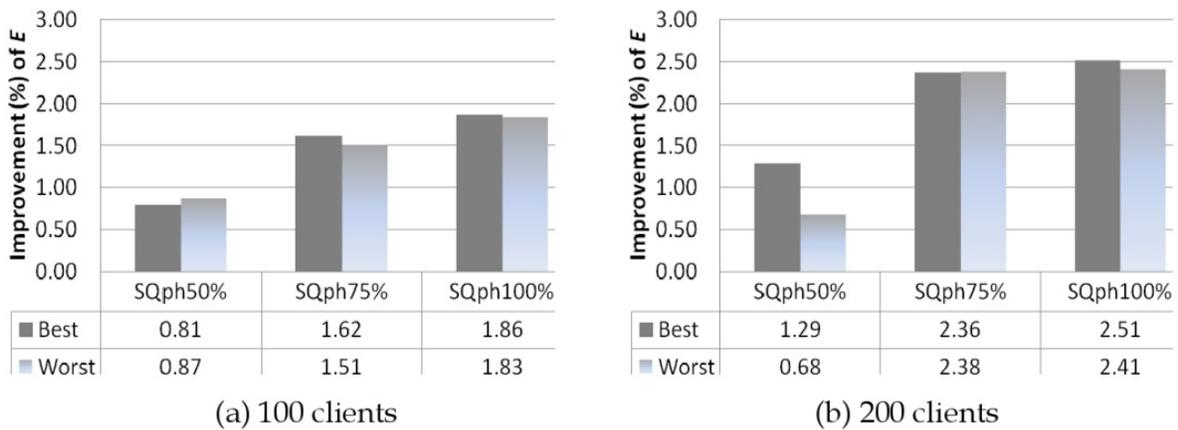


Figure 10. Improvement ratio of total traversing cost E using SQ_{ph} to that using SQ.

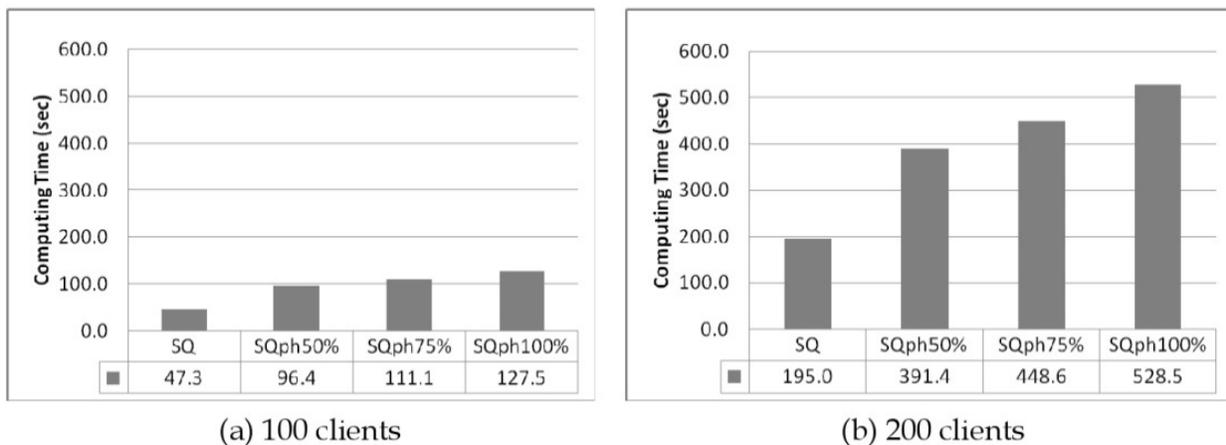


Figure 11. Computing time consumed by the methods using SQ and SQ_{ph} .

6.4. Comparison between SQ, SQ_{ph} and SQ_{ph}^*

In order to evaluate effect of device for reducing the number of vehicles in use m , results of experiments using four types of SQ_{ph}^* are compared. Processes in SQ are divided into two parts. The first half processes correspond to higher temperature, the latter processes to lower temperature. δ (coefficient of reducing pheromone on the edges connecting depot and other

clients) is set to 0.25, 0.5 and 1(= not reducing). Four types of SQ_{ph}^* (SQ_{ph}^*1 , SQ_{ph}^*2 , SQ_{ph}^*3 , SQ_{ph}^*4) are defined in Table 1.

		In the latter half processes in SQ		
		$\delta = 0.25$	$\delta = 0.5$	$\delta = 1$
In the first half processes in SQ	$\delta = 0.25$	SQ_{ph}^*3	---	SQ_{ph}^*4
	$\delta = 0.5$	---	SQ_{ph}^*1	SQ_{ph}^*2

Table 1. Four types of SQ_{ph}^* for experiments.

6.4.1. Comparison of the number of vehicles m

Computational experiments are performed ten times on all benchmark problems with 100 clients and 200 clients. Regarding the number of vehicles in use m , improvement ratio of SQ_{ph}^* to $SQ_{ph}100\%$ (also to SQ) is illustrated in Figure 12.

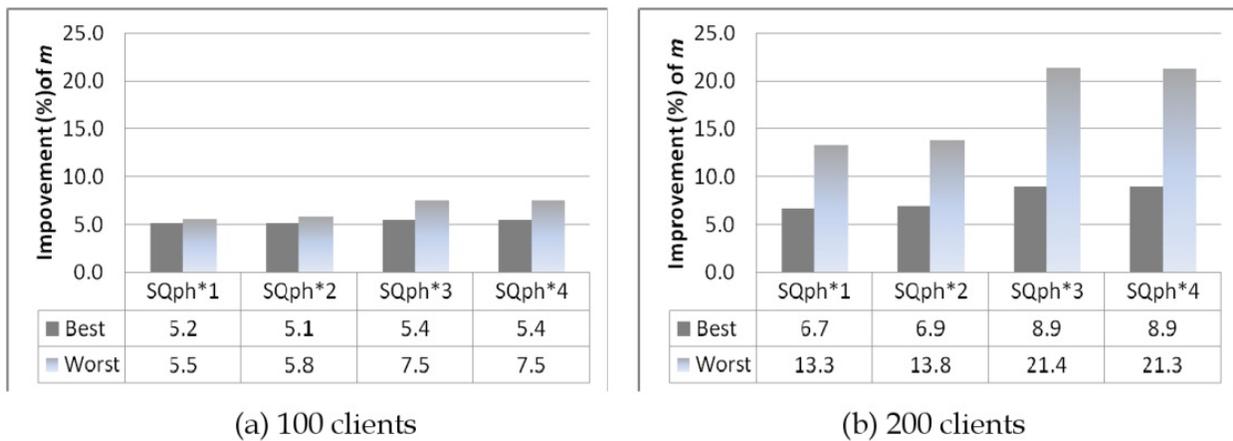


Figure 12. Improvement ratio (%) of the number of vehicles in use m using SQ_{ph}^* to that using SQ_{ph} .

According to this result, severer reducing coefficient δ (corresponding to SQ_{ph}^*3 and SQ_{ph}^*4) brings smaller number of vehicles in use. Improvement in worst cases is larger than in best cases. This situation might be caused by the fact that the value of m is so large in worst case using $SQ_{ph}100\%$ that there is potential to be greatly improved by using SQ_{ph}^* .

6.4.2. Comparison of traversing cost E

Comparison of traversing cost E is significant only between situations based on same number of vehicles m . There are 15 instances with 100 clients and 23 benchmark instances with 200 clients in which optimal m is already obtained by basic SQ. Because in these instances further reduction of m cannot be brought by SQ_{ph}^* , comparison of E is attempted in these instances. Regarding E , improvement ratio of SQ_{ph}^* to SQ is illustrated in Figure 13.

Values of E by using SQ_{ph}^*2 and SQ_{ph}^*4 are better than those by $SQ_{ph}100\%$, SQ_{ph}^*1 and SQ_{ph}^*3 . As defined in Table 1, SQ_{ph}^*2 and SQ_{ph}^*4 accompany reduction of pheromone on the edges connecting depot conducted only in the first processes in SQ_{ph} , while SQ_{ph}^*1 and SQ_{ph}^*3

accompany reduction of the pheromone in all processes in SQ_{ph} . These results are interpreted to mean that most of reduction of the number of vehicles in use is likely attained in the first processes in SQ_{ph}^* , while convergence of E mainly depends on the latter processes in SQ_{ph} . Computing time consumed by the methods using SQ , SQ_{ph} and SQ_{ph}^* is compared in Figure 14.

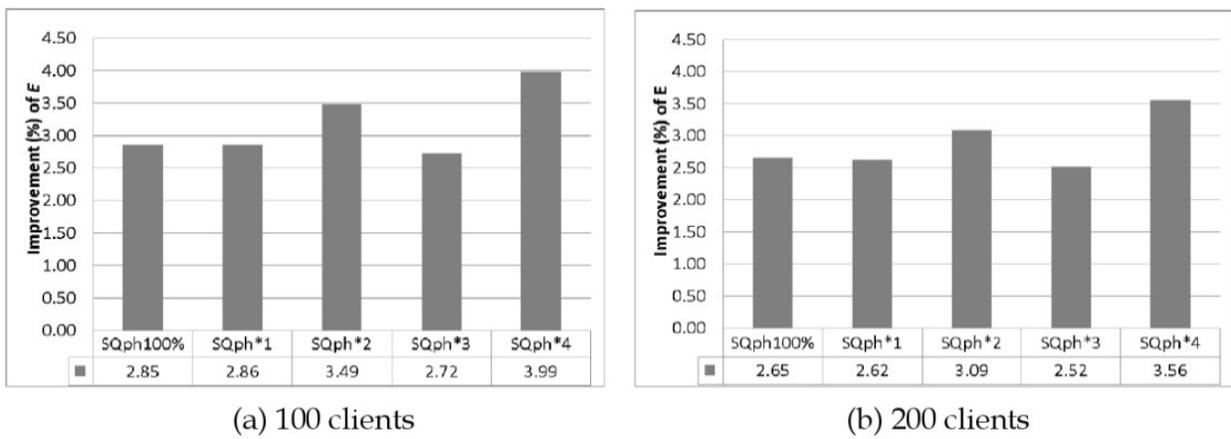


Figure 13. Improvement ratio of total traversing cost E using SQ_{ph} and SQ_{ph}^* to that using SQ .

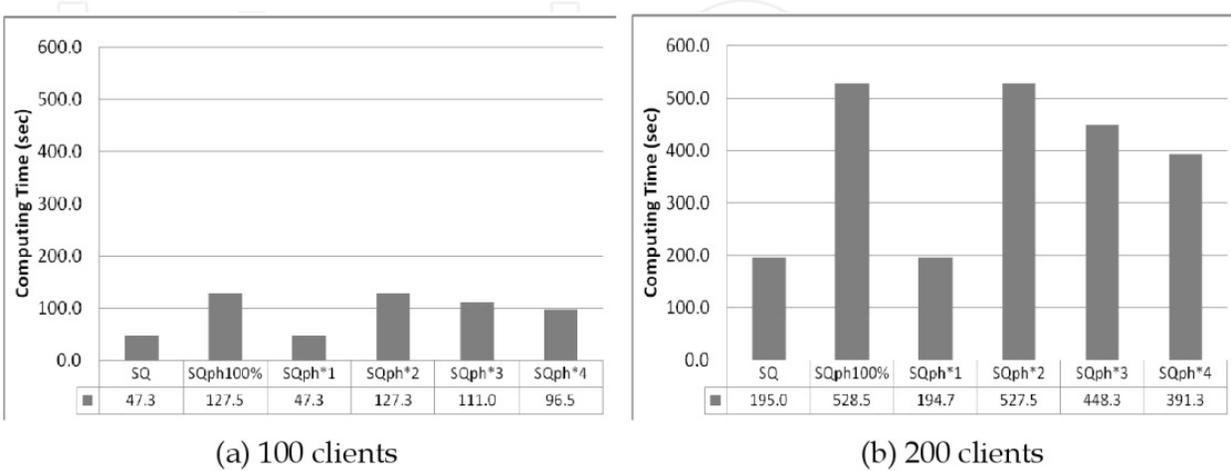


Figure 14. Computing time consumed by the methods using SQ , SQ_{ph} and SQ_{ph}^* .

To summarize these experiments, SQ_{ph}^* is the most well-balanced method among methods discussed in this chapter by taking account of both of reduction of the number of vehicles in use m and reduction of total traversing cost E . Moreover, regarding computing time, SQ_{ph}^* is moderate. Although computing time consumed by SQ_{ph}^* is about two times longer than that by basic SQ, it takes about only 1.6 min for solving 100 clients problems, and it takes about 6.5 min even in 200 clients problems. It is applicable to make actual vehicle routing plans in freight carriers.

7. Conclusion

In this chapter, in order to relieve blind searches in Simulated Quenching (SQ), that is are a practical variant of Simulated Annealing (SA), utilization good characteristics of history records during SQ search processes is attempted. Two new devices which are inspired by the effect of pheromone in ant colony optimization (ACO) are adjusted and incorporated into SQ procedures to solve VRPTW. The one is a device to reduce total traversing cost E and the other is a device to reduce the number of vehicles in use m . By computational experiments on all of 56 benchmark instances with 100 clients and all of 59 benchmark instances with 200 clients, it is shown that both of two devices are effective. However, there is a trade-off between effects for reducing E and for reducing m . Taking account of putting the right device in the right place, SQ_{ph}^* in which the device for reducing m is set in the first half processes and the device for reducing E is set in all processes in SQ seems to be the best method. Moreover, it is moderate in computing time consumed. Reducing m in the first half processes is corresponding to diversification, while reducing E in all processes is corresponding to intensification of search. Hence, it is considered that this method improves both diversification and intensification in SQ procedures.

As mentioned before, ease of implementation and flexibility are very important as well as quality of solution, running time and robustness in practical application of VRPTW algorithms to real-world problems. The proposed method is composed by a simple data model and straightforward one-phase algorithm to solve VRPTW. Therefore, the proposed method has comparative ease of implementation and much flexibility.

Two devices incorporated in SQ procedures in this chapter are able to be incorporated into SQ procedures in other routing problems which are embodied in the string model. As introduced in Sec.4.4, VRP with backhaouls (VRPB), Pickup and Delivery Problem (PDP), Capacitated Arc Routing Problem (CARP) and a general routing problem with nodes, edges, and arcs (NEARP) have already been embodied in string model and solved by SQ method. Application of two devices to these problems are left for future study.

Author details

Hisafumi Kokubugata*, Yuji Shimazaki, Shuichi Matsumoto,
Hironao Kawashima and Tatsuru Daimon
Department of Administration Engineering, Keio University, Japan

* Corresponding Author

8. References

- [1] Crescenzi, P. & Kann, V. (2000). A Compendium of NP Optimization Problem, Web site: <http://www.nada.kth.se/~viggo/wwwcompendium/node103.html>
- [2] Prins, C. & Bouchenoua, S. (2004). A Memetic Algorithm Solving the VRP, the CARP and more General Routing Problems with Nodes, Edges and Arcs, In: *Recent Advances in Memetic Algorithms, Studies in Fuzziness and Soft Computing 166*, Hart W.; Kranogor N. & Smith J. (Eds.), pp. 65-85, Springer, Berlin.
- [3] Bräysy, O. & Gendreau, M. (2005). Vehicle Routing Problem with Time Windows Part I: Route construction and local search algorithms, *Trans. Sci.*, Vol.39, No.1, pp.104–118.
- [4] Ioannou, G.; Kritikos M. & Prastacos G. (2001). A Greedy Look-ahead Heuristic for the Vehicle Routing Problem with Time Windows, *J. Oper. Res. Soc.* Vol.52, pp.523–537.
- [5] Solomon, M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, *Operations Research*, Vol. 35, No. 2, pp. 254-265.
- [6] Bräysy, O. (2003). Fast Local Searches for the Vehicle Routing Problem with Time Windows. *Inform. Systems Oper. Res.* Vol.41, pp.179–194.
- [7] Cordeau, J.-F.; Laporte G., Mercier. A. (2001). A unified tabu search heuristic for vehicle routing problems with time windows, *J. Oper. Res. Soc.* Vol.52, pp.928–936.
- [8] Homberger, J. & Gehring, H. (2005). A two-phase hybrid metaheuristic for the vehicle routing problem with time windows. *Eur. J. Oper. Res.* Vol.162, pp.220–238.
- [9] Berger, J.; Barkaoui, M. & Bräysy, O. (2003). A Route-Directed Hybrid Genetic Approach for the Vehicle Routing Problem with Time Windows, *Inform. Systems Oper. Res.*, Vol.41, pp.179–194.
- [10] Gambardella, L. M.; Taillard, E. & Agazzi, G. (1999). MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Windows. In: *New Ideas in Optimization*, Corne, D., Dorigo, M. & Glover, F. (Ed). pp. 63–76, McGraw-Hill, London, UK.
- [11] Chiang, W. C. & Russell, R. A. (1996). Simulated Annealing Metaheuristics for the Vehicle Routing Problem with Time Windows. *Ann. Oper. Res.* Vol. 63 pp.3–27.
- [12] Bräysy, O. & Gendreau, M. (2005). Vehicle Routing Problem with Time Windows Part II: Metaheuristics, *Trans. Sci.*, No.39, No.1, pp.119–139.
- [13] Bent, R. & P. Van Hentenryck. (2004). A two-stage hybrid local search for the vehicle routing problem with time windows, *Transportation Sci.*, Vol.38, No.4, pp.515–530.
- [14] Bräysy, O. (2003b). A Reactive Variable Neighborhood Search for the Vehicle Routing Problem with Time Windows, *INFORMS J. Comput.*, Vol.15, pp.347–368.
- [15] Ibaraki, T.; Imahori, S., Kubo, M., Masuda, T., Uno, T. & Yagiura, M. (2005). Effective Local Search Algorithms for Routing and Scheduling Problems with General Time Window Constraints, *Transportation Science*, Vol. 39, No. 2, pp.206-232.

- [16] Hasama, T.; Kokubugata, H. & Kawashima H. (1999). A Heuristic Approach Based on the String Model to Solve Vehicle Routing Problem with Various Conditions, *Preprint for World Congress on Intelligent Transport Systems*, No.3027, Toronto, Canada, Nov. 1999.
- [17] Gendreau, M.; Laporte, G. & Potvin, J.-Y. (2002). Metaheuristics for the Capacitated VRP, In: *The Vehicle Routing Problem*, Toth P. & Vigo, D. (Ed), pp. 129–154, SIAM, Philadelphia, USA.
- [18] Van Breedam, A. (1996). An Analysis of the Effect of Local Improvement Operators in GA and SA for the Vehicle Routing Problem, *RUCA working paper 96/14*, University of Antwerp, Belgium.
- [19] Kokubugata, H.; Itoyama, H. & Kawashima, H. (1997). Vehicle Routing Methods for City Logistics Operations, *Preprint for 8th IFAC Symposium on Transportation Systems*, pp.727-732, Hania, Greece, June 1997.
- [20] Ingber, A. L. (1993). Simulated annealing: Practice versus theory, *J Mathl. Comput. Modelling*, Vol.18, No.11, pp.29-57.
- [21] Hasama, T.; Kokubugata H. & Kawashima H. (1998). A Heuristic Approach Based on the String Model to Solve Vehicle Routing Problem with Backhauls, *Preprint for 5th Annual World Congress on Intelligent Transport Systems*, No. 3025, Seoul, Korea, Oct. 1998.
- [22] Kokubugata, H.; Hirashima, K. & Kawashima, H. (2006). A Practical Solution of Capacitated Arc Routing for City Logistics, *Proceeding of 11th IFAC Symposium on Control in Transportation Systems*, No.222.
- [23] Kokubugata, H.; Moriyama, A. & Kawashima, H. (2007). A Practical Solution Using Simulated Annealing for General Routing Problems with Nodes, Edges, and Arcs, *Lecture Notes in Computer Science*, Vol. 4638 (SLS2007), pp. 136-149, Springer, Berlin.
- [24] Kokubugata, H. & Kawashima, H. (2008). Application of Simulated Annealing to Routing Problems in City Logistics, in *Simulated Annealing*, Cher Ming Tan (Ed.), pp.131-154, I-Tech Education and Publishing, Vienna.
- [25] Dorigo, M.; Maniezzo, V. & Colorni, A. (1996).Ant System: Optimization by a Colony of Cooperating Agents. *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, Vol. 26 No.1, pp.29-41.
- [26] Solomon, M., (2005). Web site: <http://w.cba.neu.edu/~msolomon/home.htm>
- [27] Gehring, H. & Homberger J. (2001). A Parallel Two-phase Metaheuristic for Routing Problems with Time Windows, *Asia-Pacific Journal of Operational Research*, 18, 35-47.
- [28] Gehring, H. & Homberger J. (2001). Web site: <http://www.fernuni-hagen.de/WINF/touren/inhalte/probinst.htm>
- [29] Johnson, D.S.; Aragon, C.R.; MacGeoch, L.A. & Schevon, C. (1989). Optimization by Simulated Annealing : An Experimental Evaluation, Part I, Graph Partitioning, *Operations research*, Vol. 37, pp. 865-892.

- [30] Johnson, D.S.; Aragon, C.R.; MacGeoch, L.A. & Schevon, C. (1991). Optimization by Simulated Annealing : An Experimental Evaluation, Part II, Graph Colouring and Number Partitioning, *Operations research*, Vol. 39, pp. 378-406.

IntechOpen

IntechOpen