

Application of Simulated Annealing to Routing Problems in City Logistics

Hisafumi Kokubugata and Hironao Kawashima
*Department of Administration Engineering, Keio University
 Japan*

1. Introduction

The R & D activities to realize systems which provide road traffic information and route guidance have been conducted as core systems of Intelligent Transport Systems (ITS). However, the implementation of these systems will have less effect on freight transport unless logistics operation is rationalized in parallel to the development of ITS. On the other hand, according to the expansion of internet, information has been exchanged with extremely high speed and low cost. Nevertheless, goods must be moved in the real space. E-commerce has caused the increase of door-to-door deliveries. 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). The loading rate of trucks has decreased and the rate of freight transportation in total road traffic has increased. 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 the amount of CO₂. Freight transportation in urban areas that is described above is called city logistics (Taniguchi et al. 2001).

Many researches on routing problems have been appeared in the literature. Comprehensive and detailed explanations of theoretical models and solutions of them are given by Toth & Vigo (Toth & Vigo, 2002). On the other hand, in the context of city logistics, real routing problems should not be based under the assumption on the symmetry of the link costs of visiting customer j after customer i or customer i after customer j , $p_{ij} \neq p_{ji}$, and other related mathematical properties, as triangular property etc. This is due to the fact that in an urban environment routes using the streets have to account for one way streets, issues related to regulations at intersections. In addition, travel time might vary according to traffic conditions, that is to say, it might be time dependent. Moreover, in urban road networks, demands might be located on not only spots on streets but also streets themselves. This chapter is aimed for describing the original solution, which has been invented by the authors of this chapter, to routing problems in city logistics.

At the beginning of this chapter, a variety of routing problems will be introduced and followed by the explanation of features of routing problems in city logistics. And then, a practical solution method, which is composed of a data model, transformation rules of a solution on the data model and an overall algorithm using Simulated Annealing for solving

Source: Simulated Annealing, Book edited by: Cher Ming Tan, ISBN 978-953-7619-07-7, pp. 420, February 2008, I-Tech Education and Publishing, Vienna, Austria

a variety of routing problems in city logistics, is proposed in this chapter. Evaluation of the proposed method is conducted by comparisons on computational results with those derived from other heuristics.

2. Typical routing problems in city logistics

Typical routing problems are abstracted from actual logistics operations in urban areas and formalized as mathematical problems. They are categorized as the combinatorial optimization problems. In this section, according to the type of the place that demand belongs, three problems are distinguished. They are introduced as follows.

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 (Crescenzi & Kann, 2000) for their compendium of *NP* optimization problems. Although VRP is not listed in the compendium, it is given by Prins & Bouchenoua (Prins & Bouchenoua, 2004) as follows.

- **INSTANCE:** Complete undirected graph $G = (V, E)$, initial vertex $s \in V$, vehicle capacity $W \in \mathbb{N}$, length $c(e) \in \mathbb{N}$ for each $e \in E$, demand $q(i) \in \mathbb{N}$ for each $i \in V$.
- **SOLUTION:** A set of cycles (trips), each containing the initial vertex s , 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.

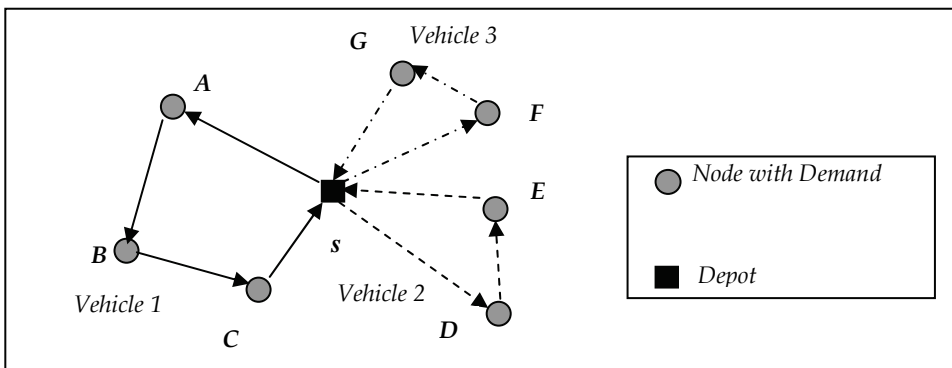


Fig. 1. Vehicle Routing Problem (VRP)

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).

2.2 Capacitated Arc Routing Problem (CARP)

When people observe deliveries in urban area, it is understood that some delivery or pickup demands belong not to spots but to streets. This circumstance contains the case where the demands are densely located along a street such as postal deliveries, and the case where the demand belongs to a street itself such as garbage collections and snow removals. In these cases, they are more suitable to be formulated as the Capacitated Arc Routing Problem (CARP) rather than as VRP. CARP was introduced by Golden & Wong (Golden & Wong, 1981). It consists of determining a set of vehicle trips at minimum total cost, such that each trip starts and ends at a depot, each required undirected edge is serviced by one single trip, and the total demand handled by any vehicle does not exceed its loading capacity. The definition of CARP is also given by Prins & Bouchenoua in the Crescenzi & Kann's style.

- **INSTANCE:** Undirected graph $G = (V, E)$, initial vertex $s \in V$, vehicle capacity $W \in \mathbb{N}$, subset $E_R \subseteq E$, length $c(e) \in \mathbb{N}$ and demand $q(e) \in \mathbb{N}$ for each edge $e \in E_R$.
- **SOLUTION:** A set of cycles (trips), each containing the initial vertex s , that collectively traverses each edge of E_R at least once. Each edge of E_R 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 comprises the costs of its traversed edges, serviced or not.

However, when the actual city logistics is considered, the original CARP is merely able to express arc routing operations in the real world imperfectly. To take waste collection as an example, there are many one-way streets in urban areas. Besides, even in two-way streets, vehicles often collect waste along one side of the street only, because broad streets are often split by central reservations. Therefore, the extended CARP introduced by Lacomme et al. (Lacomme et al., 2001) that takes account of both undirected edges and directed arcs is dealt with in this chapter.

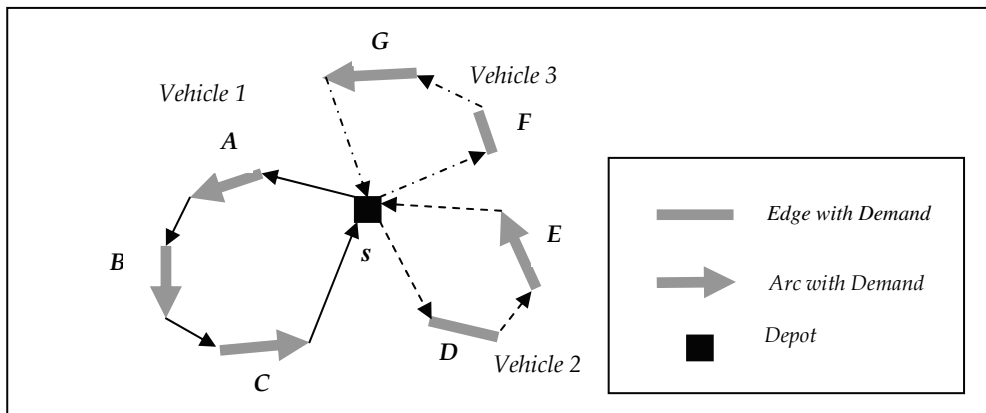


Fig. 2. The Extended Capacitated Arc Routing Problem (The Extended CARP)

2.3 General Routing Problem with Nodes, Edges, and Arcs (NEARP)

To take waste collection as an example of city logistics, there are some punctual dumps (such as factories, schools, and hospitals) that put out a large amount of waste, while other small waste dumps along a street are considered as the grouped arc demand. In order to fit the model more closely to the routing situations in the real world, Prins & Bouchenoua

defined a general routing problem with nodes, edges, and arcs (NEARP) that handles demands which belong to any of nodes, (undirected) edges and (directed) arcs (Prins & Bouchenoua, 2004).

- **INSTANCE:** Mixed graph $G = (V, E, A)$, initial vertex $s \in V$, vehicle capacity $W \in \mathbb{N}$, subset $V_R \subseteq V$, subset $E_R \subseteq E$, subset $A_R \subseteq A$, traversal cost $c(u) \in \mathbb{N}$ for each "entity" $u \in V \cup E \cup A$, demand $q(u) \in \mathbb{N}$ and processing cost $p(u) \in \mathbb{N}$ for each required entity (task) $u \in V_R \cup E_R \cup A_R$.
- **SOLUTION:** A set of cycles (trips), each containing the initial vertex s , that may traverse each entity any number of times but process each task exactly once. 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 comprises the processing costs of its serviced tasks and the traversal costs of the entities used for connecting these tasks.

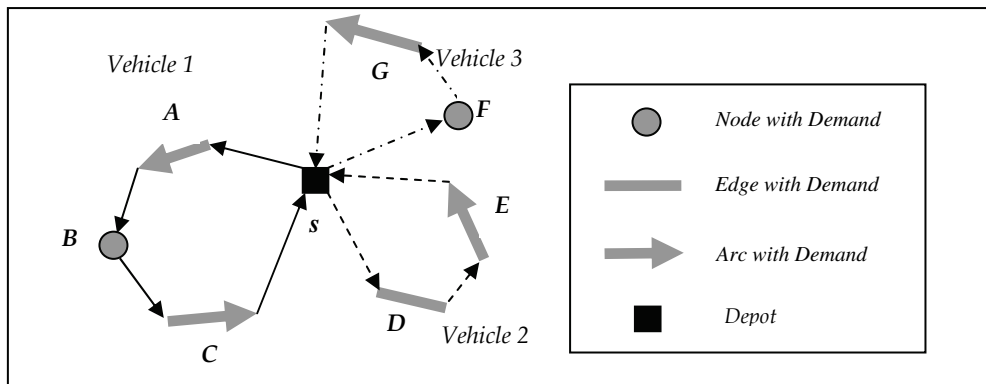


Fig. 3. Node, Edge and Arc Routing Problem (NEARP)

3. Precedent studies on heuristics for routing problems

The VRP belongs to NP -hard problems. Even concerning the simple VRP, exact methods are not fit for large problems. Therefore, heuristics have been important in the application of the VRP. Before the proposed method will be explained, precedent studies on heuristics for VRP are introduced briefly. The heuristics for solving routing problems are classified into two major classes (Toth & Vigo, 2002). One is the family of classical heuristics and the other is the family of metaheuristics including Simulated Annealing.

3.1 Classical heuristics for VRP

The heuristics belong to the first class have been specially invented for solving routing problems. They utilize the proper characteristics of routing problems. They are called classical heuristics. They are further classified into three types.

The first one is the type of constructive heuristics that produce vehicle routes by merging existing routes or inserting nodes into existing routes. The famous saving method (Clark & Wright, 1964) that made the beginning of the studies on VRP belongs to this subtype.

Chistofides, Mingozzi & Toth Insertion Heuristic (Chistofides et al., 1979) also belongs to this subtype.

The second one is the type of two-phase heuristics. Most of them assign nodes with demands to vehicles in the first phase, and then decide routing order of nodes for each vehicle in the second phase. These are called cluster-first, route-second methods. Fisher & Jaikumar Algorithm (Fisher & Jaikumar, 1981) is the typical method which belongs to this type. The optimization in the second phase which is applied to the result of optimization in the first phase is not guaranteed to derive global optimum. There are also route-first cluster-second methods which produce a giant tour including entire nodes in the first phase, and then cut and divide into vehicle routes in the second phase.

The last one is the type of improvement heuristics which make changes in one vehicle route or between several vehicle routes. Lin & Kernighan (Lin & Kernighan, 1973) method is the typical method which belongs to this type. Many methods of this type are based on λ -opt mechanism in which λ edges connecting nodes are exchanged in routes.

3.2 Metaheuristics for VRP

Metaheuristics have been introduced into the solutions for VRP 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 VRP.

Among the methods incorporating TS, Taburoute algorithm of Gendreau et al., (Gendreau et al., 1994) has had an established reputation. In each repetition in the method, one node is deleted from a vehicle route and inserted into the best position in other routes.

Among the methods incorporating GA, the method proposed by Prins (Prins, 2001) is reported to get good results. It adopts a hybrid strategy that consists of GA procedure in the giant tour without route delimiters, and local search procedures carried out in a route or between two vehicle routes.

With respect to ACO, not so many works on VRP are appeared in the literature.

Among VRP solutions using SA, the method proposed by Osman (Osman, 1993) is popular. In its main procedure, one node or two nodes are exchanged between existing two vehicle routes. The move of one node or two nodes from one vehicle route to another is also allowed.

In a comprehensive survey on metaheuristics for VRP given by Gendreau et al. (Gendreau et al., 2002), it is described that the methods based on TS are the most effective. It is also said that existing methods based on SA are not competitive with TS; while those based on GA and ACO have possibility to be competitive in future studies because they have not been fully exploited.

Most of the procedures for solving the extended routing problems are developed by making use of the procedures for VRP.

4. Data model and generating neighbours in searching process of the proposed method for VRP

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. The original solution of VRP which is composed of a simpler data model and a one phase algorithm, incorporated with original methods of generating neighbours, has been proposed by the authors of this chapter.

4.1 Data model for VRP

The model to express a state of solution of VRP is realized as a sequence of integers, i.e., a string. In the string, the position of a number, 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 Fig.4. In the string, a node with demand is expressed by a positive number. 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 number between '0' and '0', the relevant vehicle is not in use.

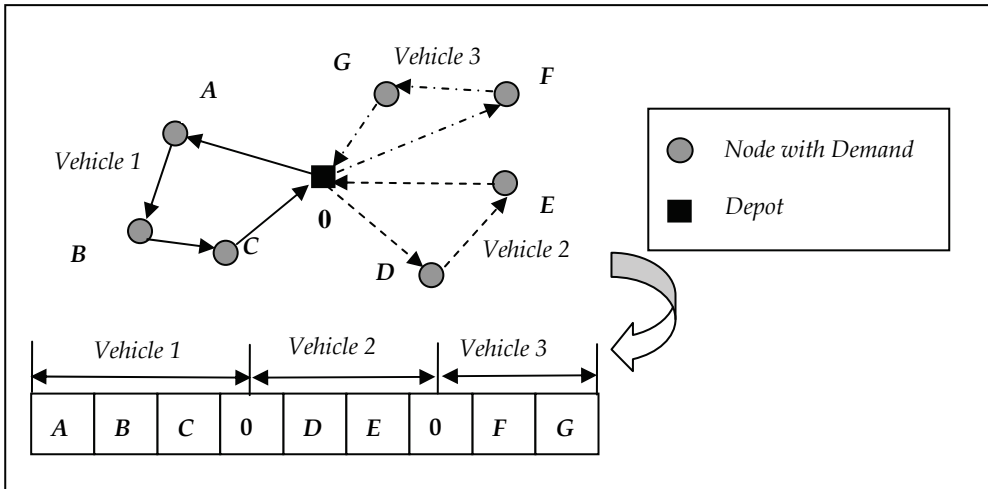


Fig. 4. Proposed Data Model for VRP

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. (Gendreau et al. 2002) as the original idea was given by Van Breedam (Van Breedam, 1996). However, the proposed transformation rules in this chapter based on the data model are quite different from those of precedent methods as they will be described in the following section.

4.2 Transformation rules for generating neighbors

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 a number with another one in the string. The second rule is to delete an arbitrary number and then insert it to another position in the string. The third rule is that after a part of the string is taken out temporally, the direction of the partial string is reversed, and then embedded in the place where the string is taken out. These three transformation rules are illustrated in Fig. 5.

Note that the rules are also applied to the special number '0' in the string data model illustrated in Fig.4. In other words, '0' is treated impartially with other numbers.

If 'one-to-one exchange' is executed within a substring partitioned by '0', only a vehicle route is changed. An example of the case is illustrated in Fig. 6. If 'one-to-one exchange' is

executed between two non-zeros striding over '0', two tasks are exchanged between two vehicle routes. An example of this case is illustrated in Fig. 7. If 'one-to-one exchange' is executed between a non-zero number and '0', two vehicle routes are merged, while another vehicle route is divided into two vehicle routes. An example is illustrated in Fig. 8.

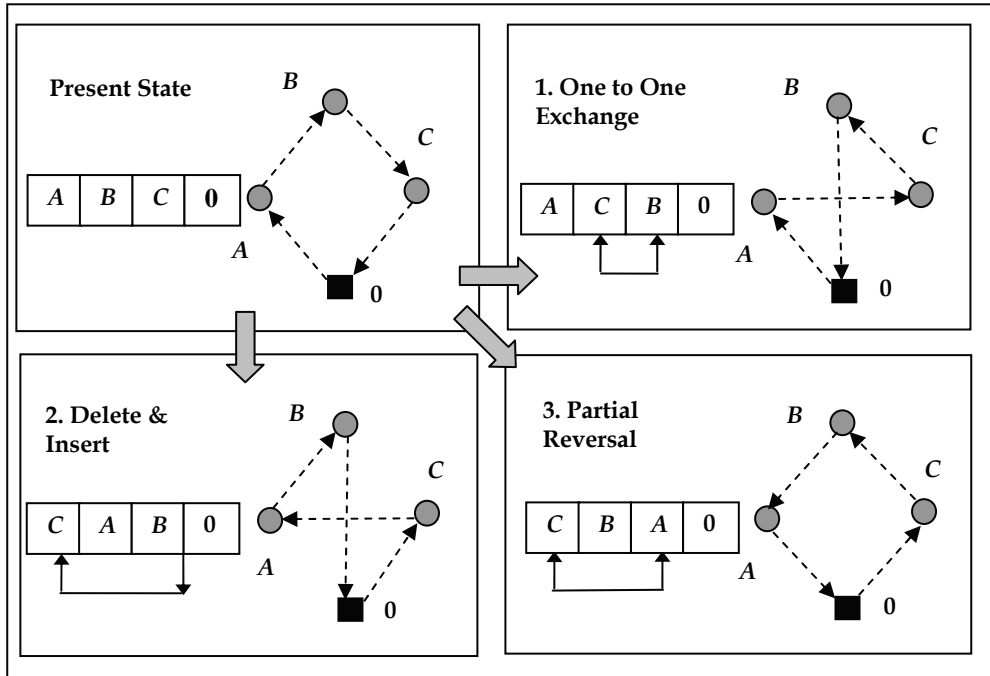


Fig. 5. Three Transformation Rules for Generating Neighbours

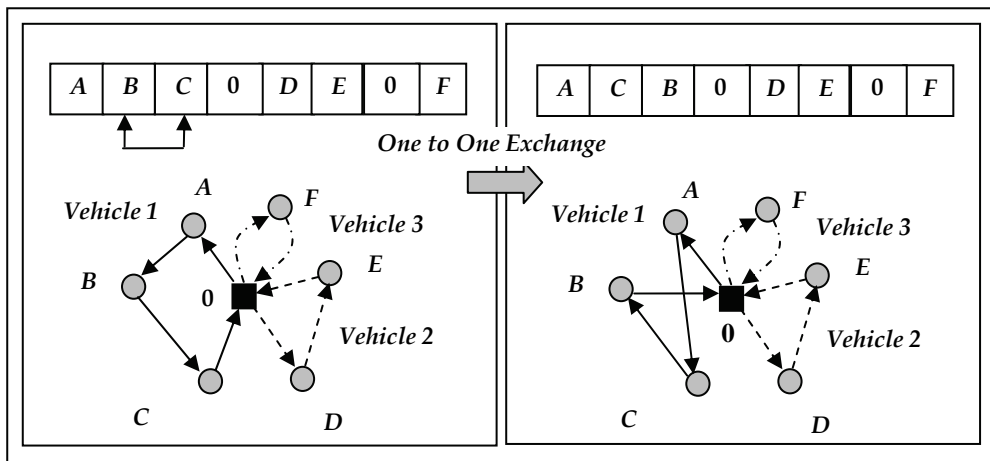


Fig. 6. A Result of 'One-to-One Exchange' within a Vehicle Route

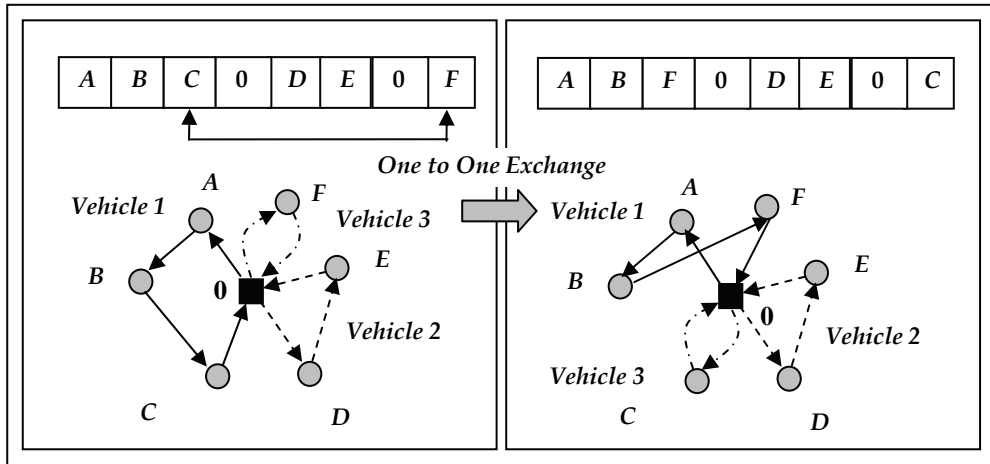


Fig. 7. A Result of ‘One-to-One Exchange’ between Two Non-Zeros Striding over ‘0’

When the second transformations rule ‘delete and insert’ is applied, several different cases also arise. If a non-zero number is deleted and inserted at ‘0’, a task is moved to another vehicle route. An example is illustrated in Fig. 9.

When the third transformations rule ‘partial reversal’ is applied, several different cases also arise. If a substring including ‘0’ is reversed, the relevant plural vehicle routes are changed. An example is illustrated in Fig.10.

These transformation rules were originally presented by the authors of this chapter (Kokubugata et al., 1997).

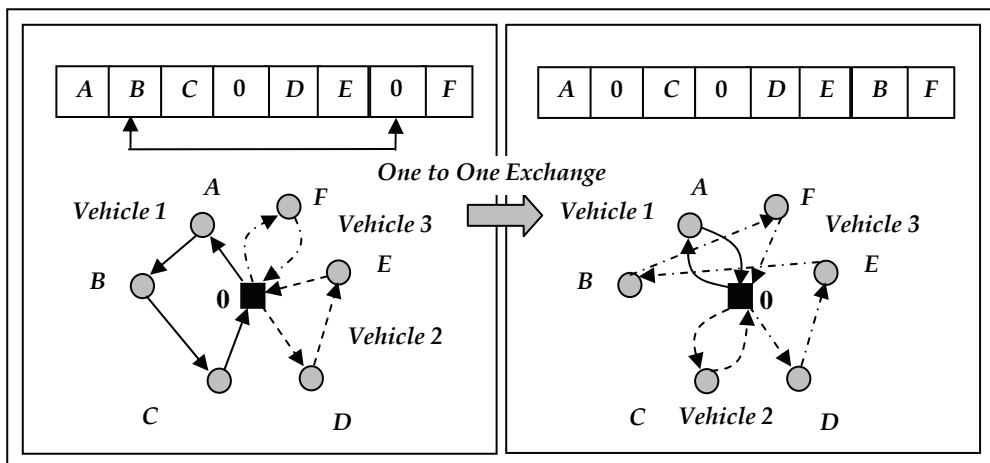


Fig. 8. A Result of ‘One-to-One Exchange’ between Non-Zero and ‘0’

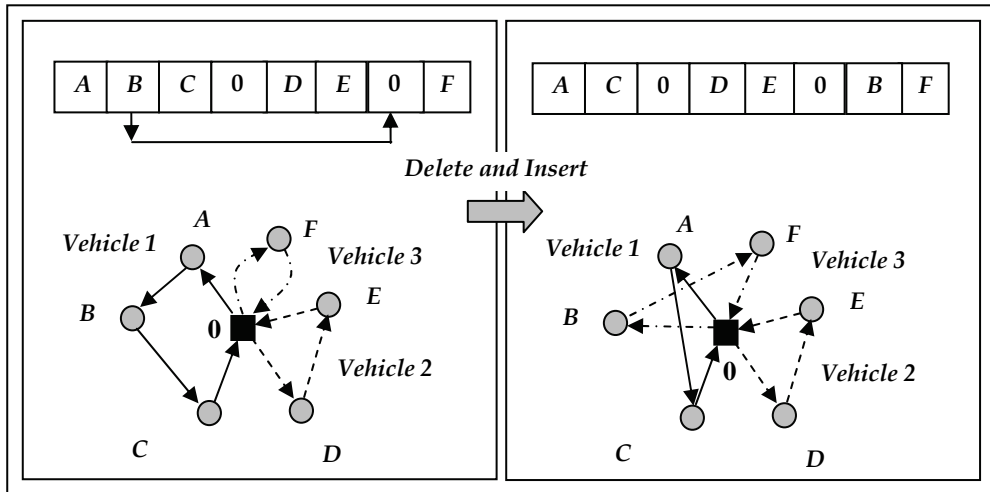


Fig. 9. A Result of Deleting Non-Zero and Inserting It at '0'

4.3 Objective function

The objective of the VRP is the minimization of total cost which is subject to constraints including the loading capacity of each vehicle. The objective function of the VRP is formulated as follows.

$$E = \sum_{i=1}^n c_{s_i} + \sum_{i=0}^n p_{s_i, s_{i+1}} \tag{1}$$

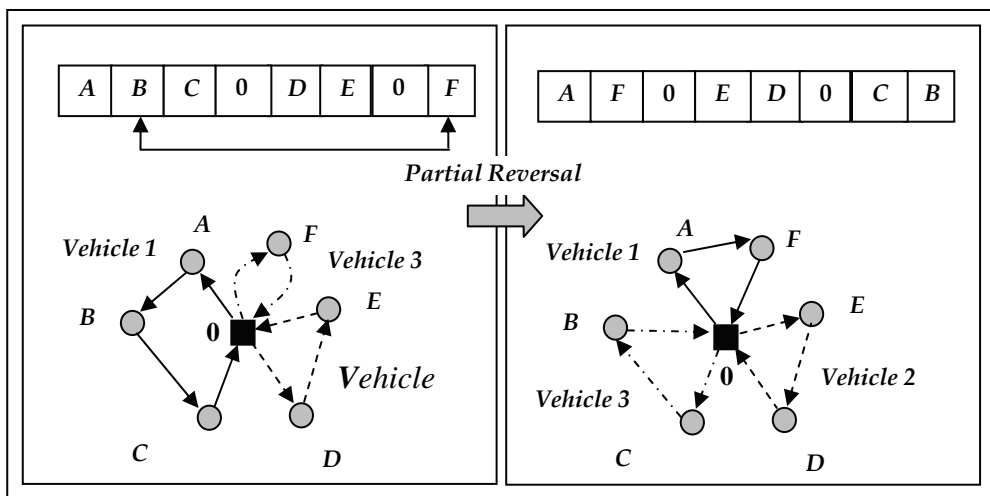


Fig. 10. A Result of 'Partial Reversal' Striding over '0'

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 cost at the node k (if $k = 0$, then $c_k = 0$); $p_{k,l}$ is the minimal traversing cost from the node k to the node l .

Each value of $p_{k,l}$ 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.

4.4 Optimization algorithm using simulated annealing

Simulated Annealing (Metropolis method) is adopted as the optimization technique for the proposed method since it is characterized by simple stochastic procedures and by global searching scope.

Starting with a random initial state, it is expected to approach an equilibrium point. 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  $p_{k,l}$ 
between all pair of tasks  $k, l$  including the depot 0;
{II. Initialization} Generate a random initial feasible solution  $x_0$ ;  $x := x_0$ ;  $x^* := x$ ;
T := INITTEMP; Set N as the averaged neighbourhood size;
{III. Optimization by SA} Minimize E by repetition of applying randomly one of the three
transformation rules to the string model corresponding to  $x$  in the framework of SA;
{IV. Output} Output the best solution  $x^*$ .
(2)
```

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  $x'$  from the current state  $x$  by applying randomly
    one of the three transformation rules to the string model of  $x$ ;
    If  $x'$  is feasible Then
      Calculate  $\Delta E = E(x') - E(x)$ ;
      If  $\Delta E < 0$  Then
         $x'$  is accepted as a new state;
        If  $E(x') < E(x^*)$  Then  $x^* := x'$ ;
      Else  $x'$  is accepted with probability  $\exp(-\Delta E/T)$ 
      If  $x'$  is accepted Then changes := changes + 1;  $x := x'$ 
    Until trials  $\geq$  SIZEFACTOR  $\cdot$  N or changes  $\geq$  CUTOFF  $\cdot$  N;
  T := T  $\cdot$  TEMPFACTOR
Until T  $\leq$  INITTEMP / FINDIVISOR
(3)
```

As sketched in Sec. 3.2, in the existing methods using metaheuristics including SA, the transformation procedure of a solution is carried out intentionally between two existing vehicle routes. However, 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.

5. Analysis of generating neighbours in searching process of the proposed method for VRP

As described in the previous section, the proposed method is based on the stochastic creation of neighbours on the string data model in searching process. In this section, the effects of transformations are classified and analyzed experimentally.

5.1 The instance of VRP for the analysis

An instance of VRP for the analysis is taken from the famous Solomon's bench mark problem sets produced by Solomon (Solomon, 1987) and provided from Solomon's own website (Solomon, 2005). c101 is chosen as an instance among them. In c101, the number of nodes is 100, the maximum number of vehicles is 25. Moreover, the amount of demand, coordinates, time window and service time are given for each node. Although these sets are provided for VRP with Time Windows (VRPTW), time window and service time in c101 are neglected for dealing with simple VRP.

5.2 Frequencies of three transformations

Frequencies of each of three transformations are counted in computational experiments. In the computations, according to the preliminary experiments and the reference to the recommended values by Johnson et al. (Johnson et al., 1989; 1991), the values of the parameters that appear in the proposed SA algorithm are set as follows.

$$\begin{aligned}
 N &= 2L^2 \quad (L : \text{length of string}) \\
 \text{SIZEFACTOR} &= 8 \\
 \text{CUTOFF} &= 0.2 \quad (\text{Repeat iterations in the same temperature } T, \\
 &\quad \text{until } (\text{trials} \geq \text{SIZEFACTOR} \cdot N \text{ or } \text{changes} \geq \text{CUTOFF} \cdot N)) \\
 \text{INITTEMP} &= 20 \quad (\text{Initial temperature}) \\
 \text{TEMPFACTOR} &= 0.95 \quad (T_{n+1} = 0.95 T_n) \\
 \text{FINDIVISOR} &= 50 \quad (\text{If } T \leq \text{INITTEMP} / \text{FINDIVISOR}, \text{ terminate the whole of the iterations.})
 \end{aligned} \tag{4}$$

Each of the three transformation rules mentioned in Sec.4.2 should be applied equally probably to produce a new feasible state of solution. However, the feasibility rates of the results generated from three transformations are not equal. Hence, in order to produce a feasible solution generated by each transformation with almost equal probability, applying rates of these three transformations at creating neighbours are adjusted, taking account of the feasibility rates computed in the preliminary experiment.

Frequencies of three transformations at creating neighbours are shown by the top bar in Fig. 11. Those in feasible solutions are shown by the second bar. Frequencies of feasible solutions bringing cost reduction are shown by the third bar. In the procedure of SA, a transformation is executed not only in the case of cost reduction but also in the case of cost increase at certain probability. Frequencies of three transformations which are really executed are shown by the fourth bar.

When the proportions of three transformations are focused on, the result of the experiment is shown as in Fig. 12. The sum of the amount of cost reduction brought by the

transformations with cost reduction is shown by the fifth bar. Moreover, the amount of cost reduction per frequencies is shown by the sixth bar.

As the result of the experiment, the fact that all of the three kinds of transformation are effective is presented.

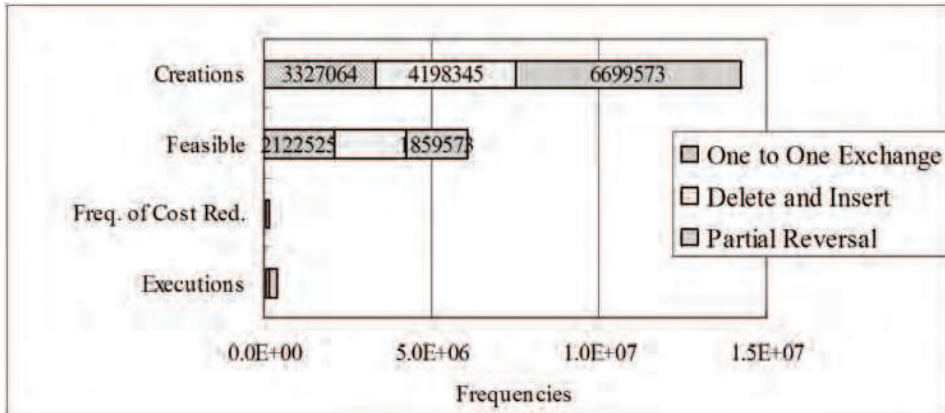


Fig.11. Frequencies of Three Transformations

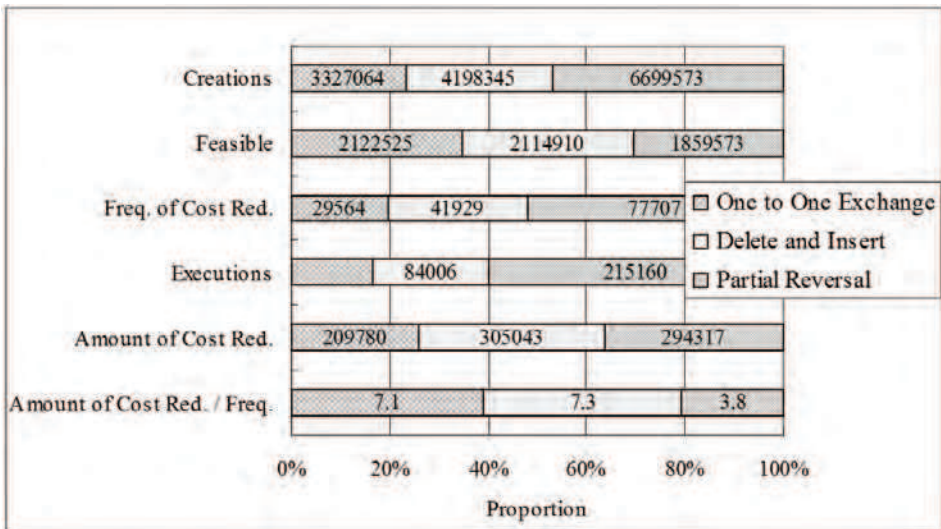


Fig.12. Proportions of Three Transformations

5.3 Classification of effects of each transformation

The core mechanism of the proposed method is based on the stochastic creation of neighbours on the string data model in searching process. Even the result of ‘one-to-one exchange’ varies as shown in Fig. 6-Fig. 8 according to the two positions selected randomly in the string data model. In this section, the effects of transformations are classified according to magnitude of the move.

5.3.1 Classification of effects of ‘One to One Exchange’

First of all, classification of effects of ‘one to one exchange’ is considered. Let p be the first selected position, and q be the second selected position in the string data model. The number s_p at p is exchanged with the number s_q at q in the string s . The effects of ‘one-to-one exchange’ are classified as follows.

- P_1 : when $p = q$. In this case, the exchange is meaningless, hence no exchange is executed.
- P_2 : when $s_p = s_q = 0$. In this case, the exchange is meaningless, hence no exchange is executed.
- P_3 : when $(s_p = 0 \ \& \ s_q \neq 0)$ or $(s_p \neq 0 \ \& \ s_q = 0)$. In this case, two vehicle routes are merged, while another vehicle route is divided into two vehicle routes; hence magnitude of the move might be large. An example of this case is illustrated in Fig.8 in Sec. 4.2.
- P_4 : when $(s_p \neq 0 \ \& \ s_q \neq 0)$ & (there is at least one ‘0’ between s_p and s_q). In this case, two nodes belonging to different vehicle routes are exchanged; hence magnitude of the move may be medium. An example of this case is illustrated in Fig.7 in Sec. 4.2.
- P_5 : when $(s_p \neq 0 \ \& \ s_q \neq 0)$ & (s_p is not adjacent to s_q) & (there is no ‘0’ between s_p and s_q). In this case, two nodes belonging to the same vehicle route are exchanged; hence magnitude of the move may be small.
- P_6 : when $(s_p \neq 0 \ \& \ s_q \neq 0)$ & (s_p is adjacent to s_q). In this case, two adjacent nodes belonging to the same vehicle route are exchanged; hence magnitude of the move may be small. An example of this case is illustrated in Fig.6 in Sec. 4.2.

A result of computational experiment is illustrated in Fig.13.

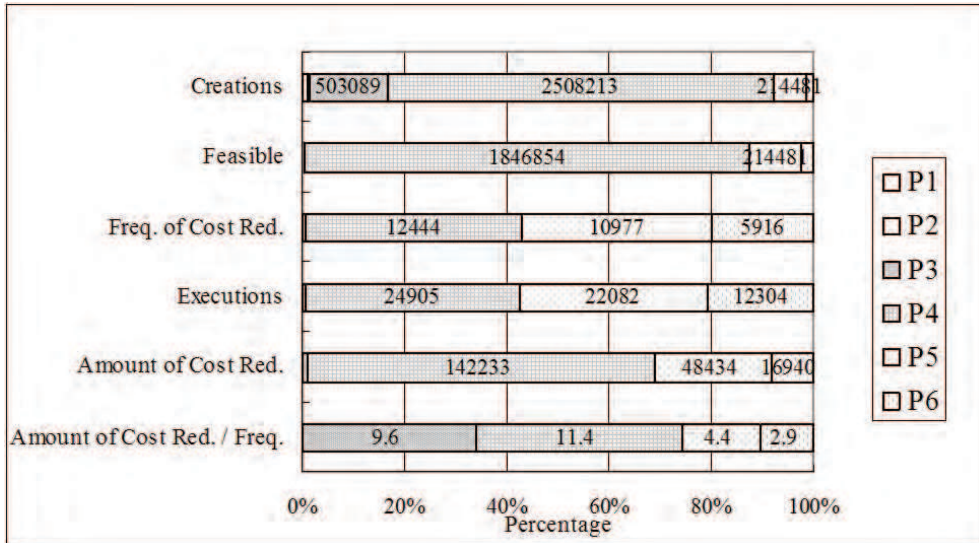


Fig.13. Effect of Each Class Related to ‘One to One Exchange’

In this figure, dark parts in the bars are expressed as the moves with large effect, while light parts as the moves with small effect. As the result of the experiment, the fact that the move with medium effect (P_4) is dominant in the SA execution is presented.

5.3.2 Classification of effects of ‘Delete and Insert’

The effects of ‘delete and insert’ are classified as follows.

- Q₁ : when $p = q$. In this case, the transformation is meaningless, hence no move is executed.
- Q₂ : when $s_p = s_q = 0$. In this case, ‘0’ is moved to the adjacent position to another ‘0’ in the string. As the result of this transformation, two vehicle routes are merged; hence magnitude of the move might be large.
- Q₃₁: when $s_p = 0$ & $s_q \neq 0$. In this case, two vehicle routes are merged; hence magnitude of the move might be large.
- Q₃₂: when $s_p \neq 0$ & $s_q = 0$. In this case, a node with demand is moved from a vehicles route to the tail or the end of another vehicle route; hence magnitude of the move may be medium. An example of this case is illustrated in Fig. 9 in Sec. 4.2.
- Q₄ : when $(s_p \neq 0$ & $s_q \neq 0)$ & (there is at least one ‘0’ between s_p and s_q). In this case, a node with demand is moved from a vehicle route into another vehicle route; hence magnitude of the move may be medium.
- Q₅ : when $(s_p \neq 0$ & $s_q \neq 0)$ & (s_p is not adjacent to s_q) & (there is no ‘0’ between s_p and s_q). In this case, a node with demand is moved to another position within the same vehicle route; hence magnitude of the move may be small.
- Q₆ : when $(s_p \neq 0$ & $s_q \neq 0)$ & (s_p is adjacent to s_q) . In this case, two adjacent nodes belonging to the same vehicle route are exchanged; hence magnitude of the move may be small.

A result of computational experiment is illustrated in Fig.14.

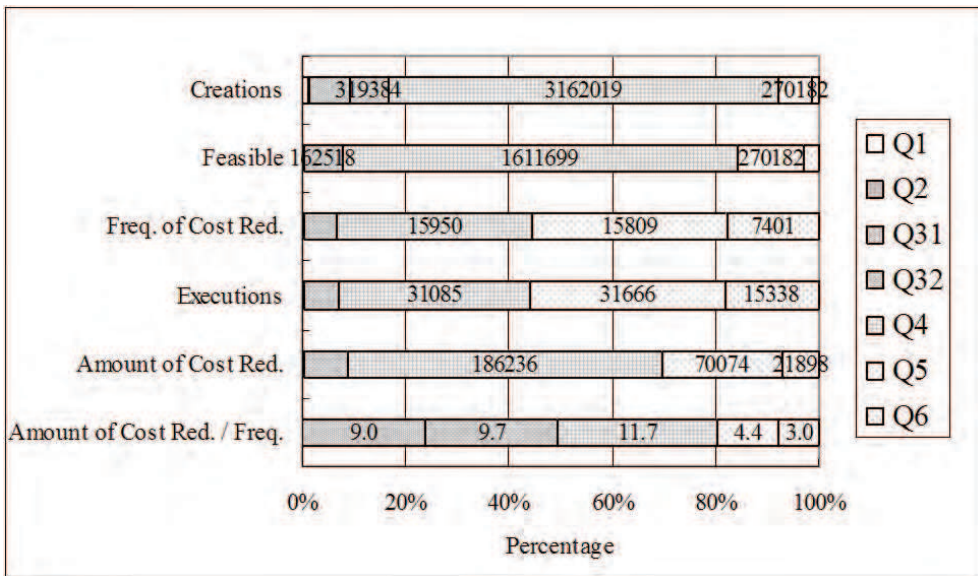


Fig.14. Effect of Each Class Related to ‘Delete and Insert’

As the result of the experiment, the fact that the move with medium effect (Q₄) is dominant in the SA execution is presented.

5.3.3 Classification of effects of 'Partial Reversal'

The effects of 'partial reversal' are classified as follows.

- R_1 : when $p = q$. In this case, the transformation is meaningless, hence no move is executed.
- R_2 : when $s_p = s_q = 0$. In this case, a substring partitioned by two '0's is reversed. As the result, vehicle routes are reversed but vehicle assignments are not changed; hence magnitude of the move may be medium.
- R_3 : when $(s_p = 0 \ \& \ s_q \neq 0)$ or $(s_p \neq 0 \ \& \ s_q = 0)$. In this case, more than one vehicle route is changed. Moreover, in the relevant vehicle routes, both composition and routing order are changed; hence magnitude of the move might be large.
- R_4 : when $(s_p \neq 0 \ \& \ s_q \neq 0)$ & (there is at least one '0' between s_p and s_q). In this case, more than one vehicle route is changed. Moreover, in the relevant vehicle routes, both composition and routing order are changed; hence magnitude of the move might be large.
- R_5 : when $(s_p \neq 0 \ \& \ s_q \neq 0)$ & $(s_p$ is not adjacent to $s_q)$ & (there is no '0' between s_p and s_q). In this case, a sub route in a vehicle route is reversed; hence magnitude of the move may be small.
- R_6 : when $(s_p \neq 0 \ \& \ s_q \neq 0)$ & $(s_p$ is adjacent to $s_q)$. In this case, two adjacent nodes belonging route to the same vehicle are exchanged; hence magnitude of the move may be small.

A result of computational experiment is illustrated in Fig.15.

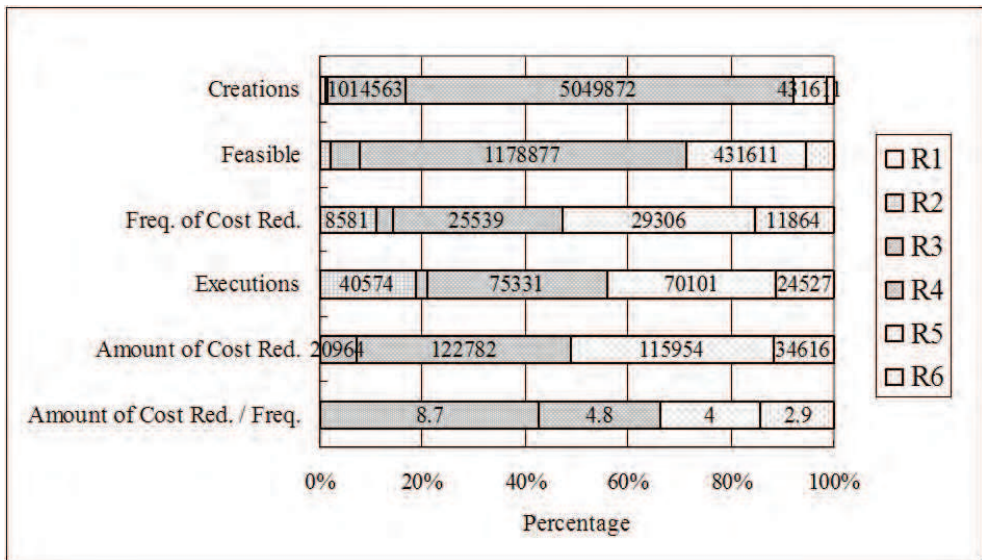


Fig. 15. Effect of Each Class Related to 'Partial Reversal'

As the result of the experiment, the fact that the move with possible large effect (R_4) and the move with small effect (R_5) are dominant in the SA procedure is presented. Through the entire observation of the effects of moves related to three transformations, it seems that each move works appropriately and achieves the expected effects in SA executions.

The tendency for the move with large effect to be dominant at higher temperature and that for the move with small effect to be dominant at lower temperature are recognized in the closer inspection over total experiments; the detailed explanation of this topic has not been presented yet by the authors of this chapter.

6. Computational experiments on the proposed method

Computational experiments have been attempted for testing the performance of the proposed method. They have been tried on typical instances for VRPTW, CARP and NEARP.

6.1 Experiments on Solomon's benchmark problems for VRPTW

In Vehicle Routing Problem with Time Windows (VRPTW), the earliest arriving time e_i and the latest arriving time l_i are specified for each client i , that is to say, the node with demand, in addition to the definition of simple VRP. Solomon's benchmark problems 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. In this chapter, 7 instances are chosen for computational experiments among 26 instances including 100 clients and 25 available vehicles. In the instance, 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 is also given to each client i , in addition to the earliest arriving time e_i and the latest arriving time l_i . The geographical data are randomly generated in problem instances R101, R102 and R108, clustered in instances C101 and C102, and a mix of random and clustered structures in instances RC101 and RC102. Because time windows are included in the constraints of the problem, objective function and the relevant procedure in the algorithm are modified in order to fit for VRPTW. Time window constraints and load capacities are treated as the penalty terms to be added to the objective function (1) as follows:

$$E = \left(\sum_{i=1}^n c_{s_i} + \sum_{i=0}^n p_{s_i, s_{i+1}} \right) + \alpha \left(\sum_{i=1}^{n+1} \max(0, a_{s_i} - l_{s_i}) \right) + \beta \left(\sum_{k=1}^m \max \left(0, \sum_{i=z_{k-1}+1}^{z_k} d_{s_i} - W_k \right) \right) \quad (5)$$

where a_{s_i} is arriving time at node s_i ; m is the number of vehicles; d_{s_i} is the amount of demand of node s_i ; z_k is the position of k th '0' in the string $s = (s_1, s_2, \dots, s_n)$ (let $z_0 = 0$; $z_m = n+1$) and W_k is the loading capacity of vehicle k . According to the modification, the position on the check of feasibility in the algorithm (3) must be changed as shown in the algorithm (6).

Repeat

$trials := 0$; $changes := 0$;

Repeat

$trials := trials + 1$;

Generate a new state x' from the current state x by applying randomly one of the three transformation rules to the string model of x ;

Calculate $\Delta E = E(x') - E(x)$;
 If $\Delta E < 0$ Then
 x' is accepted as a new state;
 If $(E(x') < E(x^*)$ and x' is feasible) Then $x^* := x'$
 Else x' is accepted with probability $\exp(-\Delta E/T)$
 If x' is accepted Then changes := changes + 1; $x := x'$
 Until trials \geq SIZEFACTOR $\cdot N$ or changes \geq CUTOFF $\cdot N$;
 T := T \cdot TEMPFACTOR
 Until T \leq INITTEMP/FINDIVISOR

In the computations, according to the preliminary experiments and the reference to the recommended values by Johnson et al. (Johnson et al., 1989, 1991), the values of the parameters that appear in the proposed SA algorithm are set as follows.

$N = 2L^2$ (L : length of string)
 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 = 20 (Initial temperature)
 TEMPFACTOR = 0.95 ($T_{n+1} = 0.95 T_n$)
 FINDIVISOR = 50 (If $T \leq$ INITTEMP / FINDIVISOR, terminate the whole of the iterations.)
 $\alpha = 20$ to 100 (to be adjusted according to the tightness of time windows), $\beta = 1$
 (In the experiment on C101 and C102, INITTEMP = 20 and FINDIVISOR = 5 are set exceptionally, because these instances are extremely easy to find the best known solutions.)

The procedure related to the creation of an initial solution in step II in the algorithm (2) should be modified in order to fit for VRPTW. The initial solution is produced by assigning nodes to vehicles in ascending order of the specified earliest arriving time; the initial solution might be infeasible.

The best solutions found by the proposed method are compared with optimum solutions obtained by exact methods and best known solutions obtained by existing heuristics. The relevant data to be compared are provided by Solomon (Solomon, 2005) and Díaz (Díaz, 2007). The computation of the proposed method is executed on Windows Vista, with Core 2 Duo, 2.0GHz CPU.

Instance	Optimum (Year Published)	Best Known by Heuristics (Year Published)	Proposed Method	Computing Time (sec)
C101	827.3 (1999)	828.94 (1999)	828.94	76
C102	827.3 (1999)	828.94 (1995)	828.94	73
R101	1637.7 (1999)	1645.79 (2000)	1644.33	194
R102	1466.6 (1999)	1486.12 (1995)	1476.12	189
R108	Not found yet	960.88 (2001)	958.98	131
RC101	1619.8 (1999)	1696.94 (1997)	1644.82	189
RC102	1457.4 (1999)	1554.75 (1997)	1480.46	188

Table 1. Computational Results on Solomon's Benchmark Problems

As shown in the Table 1, in all instances tested, the best solutions found by the proposed method are better than or equal to the best known solutions found by existing heuristics. Moreover, it seems that computing time is suited to practical use.

6.2 Experiments on instances for CARP

The proposed data model and algorithm are able to apply to the extended CARP defined in Sec. 2.2. According to the original paper by the authors of this chapter (Kokubugata et al. 2006), the outline of it is explained in this section. The data used in this method is based on an internal network coding. In the coding, entities (edges, arcs) are stored in a form which is embodied as a three dimensional array. The first component of it expresses the head node of the entity and the second expresses the tail node. The third is the Boolean value that attains 1, if and only if the entity is a directed arc. The model to express a state of solution of the extended CARP is realized as a sequence of integers, i.e., a string.

A new state of solution is generated from the present state by one of the following three types of transformation rules. The first rule is to exchange a number with another one in the string. It is the same rule as that in VRP. The second rule is to delete an arbitrary number and then insert it to another position in the string. It is also the same rule as that in VRP. However, the third rule 'partial reversion' for VRP is not adopted in the extended CARP, because 'partial reversion' likely makes infeasible neighbours in this problem. Instead, the new rule that reverse the traversing direction of an undirected edge is adopted as the third transformation rule. This rule is illustrated in Fig. 16. Of course, 'direction reversal' can not be applied to directed arcs.

Note that three rules are also applied to the special number '0' in the method for the extended CARP.

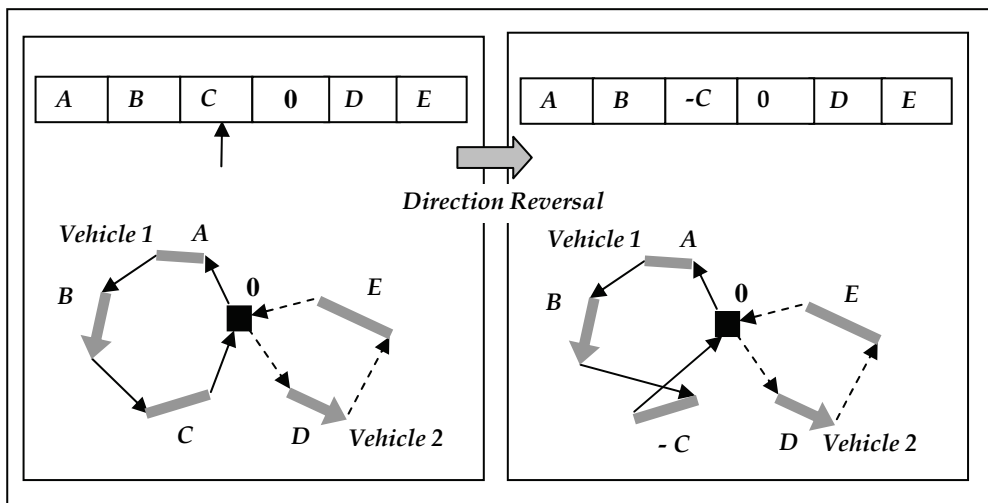


Fig. 16. A Result of 'Direction Reversal'

Instances for CARP can be obtained from the web site named CARPLIB that is supported by Belenguer (Belenguer, 2005). In the CARPLIB, some series of CARP are provided. The GDB series includes 21 small size problems, each of which contains 19-55 undirected required

arcs. The BCCM series includes 34 medium size problems, each of which contains 39-97 undirected required arcs. Link cost in these instances is not given by Euclidean distance between a pair of clients. Cost of both arc with demand and arc without demand is given directly from input data. (Edges are treated as bidirectional arcs.) Therefore, link cost between the tail of demand arc i and the head of demand arc j is obtained by calculating the shortest path traversing intermediate arcs (with demand or without demand) connecting arc i and arc j using Warshall-Floyd's algorithm.

Computational experiments are attempted to compare the proposed method with two existing heuristics. The method of Hertz et al. (Hertz et al., 2000) is based on Tabu Search, which is extended from Tabu Search used for VRP solution by Gendreau et al. (Gendreau et al., 1994) introduced in Sec. 3.2. The method of Lacomme et al. (Lacomme et al., 2001) is based on Genetic Algorithm, which is extended from Genetic Algorithm used for VRP solution by Prins (Prins, 2001) introduced in Sec. 3.2. The computation of the proposed method is executed on Windows XP, with Pentium IV, 1.8GHz CPU. In the computations, according to the preliminary experiments and the reference to the recommended values by Johnson et al. (Johnson et al., 1989; 1991), the values of the parameters that appear in the proposed SA algorithm are set as follows

$$N = 2L^2 \quad (L : \text{length of string})$$

$$\text{SIZEFACTOR} = 4$$

$$\text{CUTOFF} = 0.1 \quad (\text{Repeat iterations in the same temperature } T, \\ \text{until } (\text{trials} \geq \text{SIZEFACTOR} \cdot N \text{ or } \text{changes} \geq \text{CUTOFF} \cdot N))$$

$$\text{INITPROB} = 0.4 \quad (\text{Initial acceptance probability}) \quad (8)$$

$$\text{TEMPFACTOR} = 0.99 \quad (T_{n+1} = 0.99 T_n)$$

$$\text{FINDIVISOR} = 10 \quad (\text{If } T \leq \text{INITTEMP} / \text{FINDIVISOR}, \text{ terminate the whole of the iterations.})$$

INITTEMP must be calculated by exploratory SA executions, so as to make $\text{changes/trials} = \text{INITPROB}$ ($= 0.4$).

Instance number	Num. of vehicles	Num. of Required arcs	Hertz <i>et al.</i>	Lacomme <i>et al.</i>	Proposed Method	Computing Time (sec)
GDB1	5	22	316	316	316	3.4
GDB3	5	22	275	275	275	3.3
GDB9	10	51	317	303	309	29.6
GDB23	10	55	235	235	233	59.9
BCCM1A	2	39	173	173	173	6.6
BCCM3B	3	35	87	87	87	15.1
BCCM6C	10	50	329	317	317	22.1
BCCM9B	4	92	329	326	326	55.6

Table 2. Computational Results on CARP Instances

As shown in the Table 2, the proposed method has performance almost equal to the existing two heuristics. Moreover, it seems that computing time is satisfactorily short for the practical use.

6.3 Experiments on instances for NEARP

The proposed data model and algorithm are also able to apply to NEARP defined in Sec. 2.3. According to the original paper by the authors of this chapter (Kokubugata et al. 2007), the outline of it is explained in this section. The data used in this method is based on an internal network coding as same as that used for the extended CARP. In the coding, all entities (nodes, edges, arcs) are stored in a common form which is embodied as a three dimensional array. The first component of it expresses the head node of the entity and the second expresses the tail node. The third is the Boolean value that attains 1, if and only if the entity is an arc. If the head and the tail are the same node, the entity is understood as a single node. The model to express a state of solution of NEARP is also realized as a sequence of integers, i.e., a string.

A new state of solution is generated from the present state by one of three types of transformation rules as same as those used for the extended CARP. These are 'one to one exchange', 'delete and insert' and 'direction reversal'. Note that three rules are also applied to the special number '0' in the method for NEARP.

Prins & Bouchenoua have provided 23 instances of NEARP (Prins & Bouchenoua, 2004). These instances were produced by their original generator accompanied with randomization. They include 1-93 required nodes, 0-94 required edges and 0-149 required arcs, among 11-150 nodes and 71-311 links (integrated alias with edges and arcs). As mentioned by them, the lower bounds have not been found for their NEARP instances. The data files of NEARP instances were sent by them at the authors' request.

In the computations, all the same parameter values as used for the extended CARP (8) are used again.

Comparison between the solutions generated by the proposed method and the solutions given by Prins & Bouchenoua are conducted for 23 NEARP instances. In Table 3, the average deviations over the best value and the averaged computing time are shown. In the column of *MA*, the result of Memetic algorithm given by Prins & Bouchenoua (Prins & Bouchenoua, 2004) is shown. In the column of *Best*MA*, the performance of the best solution found with various parameter settings during their experiments is shown. *Avg₁₀SA* is the averaged result of ten computations, while *Best₁₀SA* is the best result of them. Note that the *Best₁₀SA* is obtained by computations with the standard parameter setting and it is quite different from the *Best*MA* in spite of using the same word 'Best'.

The computation of the proposed method was executed on Windows XP, with Pentium IV, 1.8GHz CPU, while the computation by Prins & Bouchenoua was executed on Windows 98, with Pentium III, 1GHz CPU.

	<i>MA</i>	<i>MATime</i> (sec)	<i>Best*MA</i>	<i>Avg₁₀SA</i>	<i>SATime</i> (sec)	<i>Best₁₀SA</i>
<i>Average</i>	1.65%	452.9 (s)	0.38%	1.51%	176.6 (s)	0.22%

Table 3. Averaged Computational Results on NEARP Instances

As shown in Table. 3, *Avg₁₀SA* is superior to *MA*, and *Best₁₀SA* obtains better results than *Best*MA*.

As a result, it is shown that the proposed method has good performance on NEARP.

7. Applications to varieties of routing problems

The proposed method is adaptable to varieties of routing problems abstracted from actual city logistics operations. In this section, three examples of varieties are explained briefly.

7.1 Routing problem with repeated trips

The case in which repetitive trips of a vehicle are allowed is frequently appeared in actual city logistics operations. After the first trip returns to the depot, unloading and loading are operated at the depot. Then, the second trip starts. This problem is also dealt with by the proposed method. In order to cope with it, another delimiter, for example '999', is introduced in the string model (Fig.17). In the transformation procedure of a solution, '999' is treated evenly with '0' as well as other numbers which represent nodes with demand.

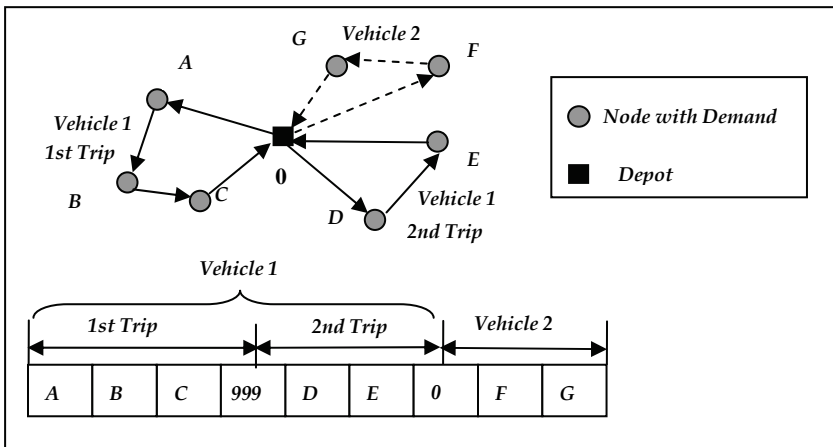


Fig. 17. VRP with Repeated Trips

This method was presented by the authors of this chapter (Hasama et al., 1999). It can be applied to the extended CARP and NEARP in the same way as VRP (Kokubugata et al., 2007).

7.2 Routing problem with plural depots

Planning of vehicle routing related to vehicles belonging to plural depots is required in actual city logistics operations. Routing problem in which plural depots are managed at a time could be also dealt with by the proposed method. The other delimiter, for example '-999', is introduced in the string model (Fig.18). In the transformation procedure of a solution, '-999' is treated evenly with '0' and other numbers.

This method was also presented by the authors of this chapter (Hasama et al., 1999). It can be applied to the extended CARP and NEARP in the same way as VRP (Kokubugata et al., 2007).

7.3 Routing problem with backhauls

In some cases of the freight transport operations, goods are delivered from the depot and empty pallets are retrieved to the depot.

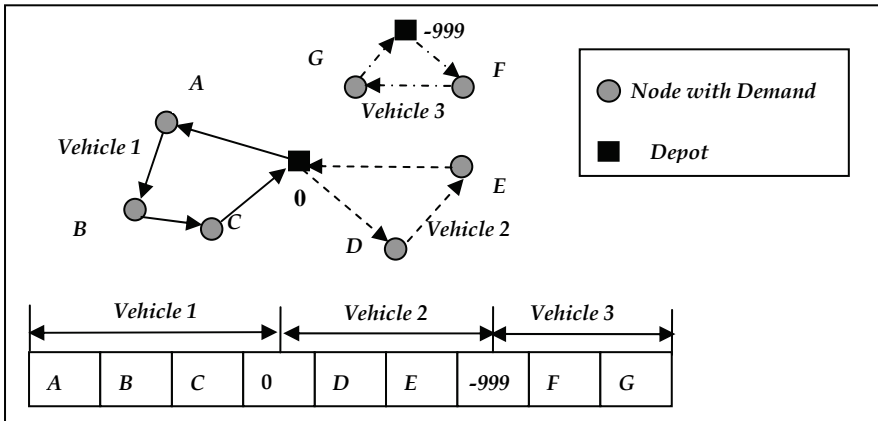


Fig. 18. VRP with Plural Depots

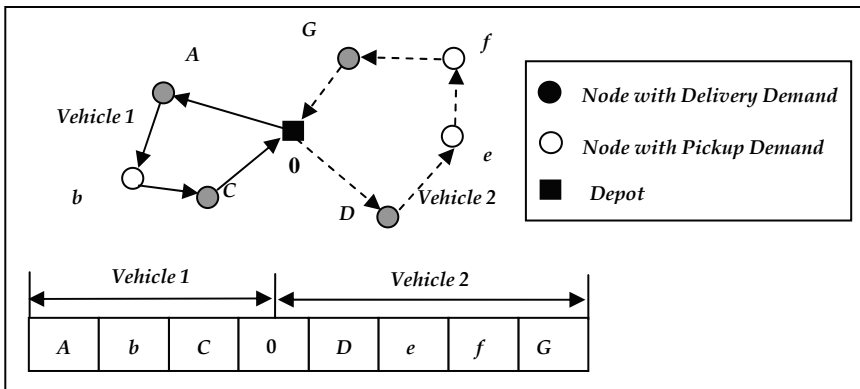


Fig. 19. VRP with Backhauls

In home-delivery service operations, both pickup and delivery services are carried out. A problem related to both deliveries and pickups is considered. Vehicles are loaded with goods at a central depot in order to service the delivery points. New goods are collected at the pickup points and brought back to the depot (backhauls). This type of problem is called the vehicle routing problem with backhauls (VRPB). The proposed method can be applied to VRPB.

In the string model corresponding to this problem, positive numbers are used for expressing delivery points, while negative numbers are used for expressing pickup points. In Fig.19, positive numbers are indicated by capital letters, while negative numbers are indicated by small letters. Each letter including '0' is treated evenly in the transformation procedure of a solution. This method was presented by the authors of this chapter (Hasama et al., 1998).

8. Conclusion

As introduced in Sec. 3.2, the applications of Simulated Annealing (SA) to routing problems have been evaluated not highly in the literature as compared with those of Tabu Search (TS) and Genetic Algorithm (GA) etc. In the core procedure of the prominent method making use of SA, only one or two nodes are exchanged between existing two vehicle routes at a time.

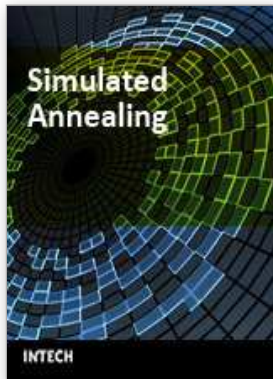
The descriptions written in this chapter have revealed that the proposed method making use of SA for routing problems has superior to other method based on other metaheuristics. The explanations made in this chapter are summarized as follows.

- The proposed method for solving the routing problems consists of simple transformation procedures applied over the entire string data model. In the framework of SA, each random application of one of transformation rules may cause the exchange of tasks between two trips, the move of a set of tasks from one trip to another trip, the exchange of tasks in the same trip, the changes in routing order in some trips and so on. Because transitions of a solution in the string data model may occasionally cause drastic changes in solutions, fast convergence to an equilibrium point might be achieved.
- The solutions generated by the proposed method are compared with the solutions given by other methods by making computational experiments on VRPTW, CARP and NEARP instances. In most cases, the proposed method shows superior performance to other methods.
- The proposed method is adaptable varieties of routing problems abstracted from practical logistics operations. The case in which repetitive trips of a vehicle are allowed, the case in which plural depots are managed at a time and routing problem with backhaul are dealt with.
- Although the proposed method is advantageous to complicated logistics operations, the following topics should be considered in order to apply the method to practical use in city logistics.
 - Applications of TS and GA to the proposed string model and the transformation rules should be attempted to compare with the proposed method making use of SA. The supposition that SA is the fittest for the string model and the transformation rules should be confirmed.
 - The proposed data model can be applied to complicated problems such as NEARP with time windows, multiple depots cases and pickup and delivery cases. The application to these cases should be examined. However, the necessary data of actual delivery cases have not been obtained yet.
 - Actual travel time may vary according to traffic conditions. Dynamic routing planning system taking account of time dependent link cost should be studied.

9. References

- Belenguer, J.M. (2005). Web site: <http://www.uv.es/~belengue/carp/>
- Clarke, G. & Wright, W. (1964). Scheduling of Vehicles from a Central Depot to a Number of Delivery Points, *Operations research*, Vol. 12, pp. 564-581
- Chistofides, N.; Mingozzi, A. & Toth, P. (1979). The Vehicle Routing Problem, In: *Combinatorial Optimization*, Chistofides, N. ; Mingozzi, A. ; Toth, P. & Sandi, C. (Eds.), pp. 315-338, Wiley, Chichester
- Crescenzi, P. & Kann, V. (2000). A Compendium of NP Optimization Problem, Web site: <http://www.nada.kth.se/~viggo/wwwcompendium/node103.html>
- Díaz, B. D. (2007). Web site: <http://neo.lcc.uma.es/radi-aeb/WebVRP/index.html>
- Fisher, M. L. & Jaikumar, R. (1981). A Generalized Assignment Heuristic for Vehicle Routing, *Networks*, Vol. 11, pp. 109--124.
- Gendreau, M.; Hertz, A. & Laporte, G. (1994). A Tabu Search Heuristic for the Vehicle Routing Problem, *Management Science*, Vol. 40, pp. 1276-1290.
- 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,

- Golden, B.L. & Wong, R.T. (1981). Capacitated Arc Routing Problems, *Networks*, Vol. 11, pp.305–315
- 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
- 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
- Hertz, A.; Laporte, G. & Mittaz, M. (2000). A Tabu Search Heuristic for the Capacitated Arc Routing Problem, *Operations research*, Vol. 48, pp. 129–135
- 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
- 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
- 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
- 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, Delft, The Netherlands, Aug. 2006
- 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
- Lacomme, P.; Prins, C. & Ramdane-Cherif, W. (2001). A Genetic Algorithm for the Capacitated Arc Routing Problem and Its Extensions, *Lecture Notes in Computer Science*, Vol. 2037, pp. 473–483, Springer, Berlin
- Lin, S. & Kernighan, B. W. (1973). An Effective Heuristic Algorithm for the Traveling Salesman Problem, *Operations research*, Vol. 21, pp. 498-516.
- Osman, I. H. (1993). Metastrategy Simulated Annealing and Tabu Search Algorithms for the Vehicle Routing Problem, *Annals of Operations Research*, Vol. 41, pp. 421-451
- Prins, C. (2001). A Simple and Effective Evolutionary Algorithm for the Vehicle Routing Problem, *Research Report*, University of Technology of Troyes, France, *Computers & operations research*, 2004, Vol. 31, No. 12, pp. 1985-2002.
- 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
- Solomon, M. (1987). Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints, *Operations Research*, Vol. 35, No. 2, pp. 254-265
- Solomon, M., (2005). Web site: <http://w.cba.neu.edu/~msolomon/home.htm>
- Taniguchi, E.; Thompson R.G.; Yamada T. & Van Duin R. (2001). *City Logistics: Network Modelling and Intelligent Transport Systems*, Pergamon, Oxford
- Toth, P. & Vigo, D. (Eds.) (2002). *The Vehicle Routing Problem*, SIAM, Philadelphia
- 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



Simulated Annealing

Edited by Cher Ming Tan

ISBN 978-953-7619-07-7

Hard cover, 420 pages

Publisher InTech

Published online 01, September, 2008

Published in print edition September, 2008

This book provides the readers with the knowledge of Simulated Annealing and its vast applications in the various branches of engineering. We encourage readers to explore the application of Simulated Annealing in their work for the task of optimization.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Hisafumi Kokubugata and Hironao Kawashima (2008). Application of Simulated Annealing to Routing Problems in City Logistics, Simulated Annealing, Cher Ming Tan (Ed.), ISBN: 978-953-7619-07-7, InTech, Available from: http://www.intechopen.com/books/simulated_annealing/application_of_simulated_annealing_to_routing_problems_in_city_logistics

INTECH

open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2008 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.