

# Knowledge-embedded Multi-stage Genetic Algorithm for Interactively Optimizing a Large-scale Distribution Network

Takashi Onoyama\*, Sen Kubota\*, Kazuko Oyanagi\* and Setsuo Tsuruta\*\*

\* Research & Development Department  
Hitachi Software Engineering Co., Ltd.  
6-81, Onoe-cho, Naka-ku, Yokohama, 231-0015, Japan  
e-mail: {onoyama, kubosen, oyanagi}@ori.hitachi-sk.co.jp  
\*\* Systems Development Laboratory, Hitachi, Ltd.  
1099 Ohzenji Asao-ku Kawasaki-shi, 215-0013, Japan  
e-mail: tsuruta@sdl.hitachi.co.jp

## Abstract

To optimize large-scale distribution networks, solving about 1000 middle scale (around 40 cities) TSPs (Traveling Salesman Problems) within an interactive length of time (max. 30 seconds) is required. Yet, expert-level (less than 3%) accuracy is necessary. To realize the above requirements, a knowledge-embedded multi-stage GA method was developed. This method combines a high-speed GA with a knowledge-embedded GA having problem-oriented knowledge effective for some special location patterns. When conventional methods were applied, solutions for more than 20 cases out of 20000 cases were below expert-level accuracy. But the developed method could solve all of 20000 cases at expert-level.

## 1. Introduction

The efficiency of products distribution remains on a lower level in Japan than in the U.S. compared to the productivity of manufacturers. This inefficiency causes economical and social problems such as necessity of the curtailment of transport expenses and environmental requirement for urgently reducing the volume of auto exhausts in Japan. In order to improve the distribution efficiency, for instance, we are aiming at optimizations of parts supply/distribution across multiple enterprises. To simulate and optimize such parts supply/distribution across multiple enterprises, the followings are considered to be necessary.

First, the conditions have to be manually set up, concerning locations of more than ten factories (parts integration points for production), locations of dozens of deposits (intermediate depositories/storehouses of parts), and allocation of trucks to transport parts. Then, for each set of above-mentioned conditions, it is necessary to automatically create several hundreds of supply/distribution routes among several hundreds of parts makers (suppliers), and tens of deposits and factories. Finally, it is required to calculate their total costs and to globally evaluate these outputs.

To globally evaluate these outputs, human judgement is indispensable and interactive response time (less than tens of seconds) is required. Thus, the system needs to create about 1000 or several hundreds of distribution routes within at least tens of seconds, therefore, 1 route has to be

created within tens of milliseconds. Since the creation of each route is equivalent to a TSP (Traveling Salesman Problem) of tens of (max 40) cities, approximate solving methods are required to ensure the response time necessary for the above human interaction.

Now, solutions generated by domain experts may have 2~3% of deviation from the mathematical optimal solution, but they never generate worse solutions which may cause practical problems. On the other hand, conventional approximate solving methods (Yamamoto and Kubo 1997) may more often generate a mathematically optimal solution, but they cannot ensure the amount of errors below 2~3%. The latter possibly causes serious problems. Thus, those conventional methods were not practically useful, especially for the above-mentioned applications.

Strict TSP solving methods such as branch and cut method and Dynamic Programming (DP) or approximate solving methods using Simulated Annealing (SA) and tabu search (Ibaraki 1993; Hooker and Nataraj 1995) take much time for calculation. Therefore, they cannot guarantee the above-mentioned responsiveness necessary for interactive simulations. Lin-Kernighan (LK) method and its improved version (Lin and Kernighan 1972) are also proposed as solving methods of the TSP. However, they cannot constantly guarantee expert-level accuracy.

Thus, the authors of this paper developed a method which efficiently solves the TSP, using GA. This method enables to guarantee the responsiveness through limiting the number of generations of GA and through improving genetic operations (initial generations, mutation and crossover) (Onoyama et al. 2000). However, in some distribution patterns, this solving method failed into local minimum and could not achieve expert-level accuracy. Therefore, we needed to further improve the solving method to guarantee expert-level accuracy always.

In the second section, the simulator for the parts supply/distribution network evaluation and its technical problems are described. In the third section, the method for solving the problem is proposed. Then, in the fourth section, experiments to validate its effect and its results are shown. In the fifth section, the effectiveness of the solving method will be proved based on the experiment, and in the sixth section, we will compare it with other methods. And in the

seventh section, the result will be concluded.

## 2. Problems in Large-scale Distribution Network Simulation

### 2.1 Large-scale Distribution Network Simulation

A distribution network across multiple manufacturing enterprises is shown in figure 1. Parts for production are delivered from parts makers (suppliers) directly to factories or through deposits. Parts are not delivered to a factory or a deposit independently by each parts maker, but a truck goes around several parts makers and collects parts. This improves distribution efficiency, which contributes to the curtailment of distribution expenses and to the reduction of the volume of auto exhausts.

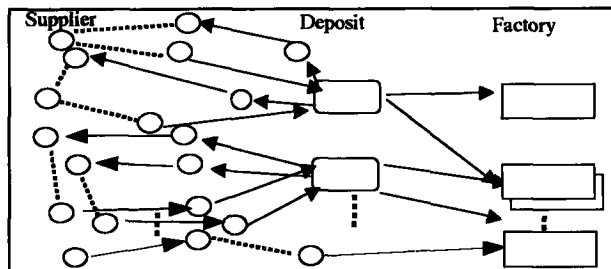


Figure 1. Large-scale Distribution Network

- 1) Set Conditions
  - i) Setting Factory Locations
  - ii) Setting Depository Locations
  - iii) Setting Parts Maker (Supplier) Locations
- 2) Create delivery routes and calculate costs
- 3) Decide Optimal Distribution Network

Figure 2. Distribution Network Simulation Process

In optimizing the above-mentioned large-scale distribution logistic network, we need to grasp the total cost of distribution under various conditions by repeating the simulation process as shown in figure 2. To calculate distribution cost in each simulation, it is necessary to create delivery routes. However, there are several hundreds of parts makers (suppliers), dozens of deposits and more than ten factories. Therefore, there are about 1000 distributing routes each of which goes around dozens (max. 40) of parts makers starting from one of the deposits or factories. Thus, in each simulation, delivery route creation is repeated about 1000 times for a set of conditions manually set up, the total delivery cost is calculated, and a person in charge globally decides the network optimality as shown in figure 2.

### 2.2 Technical Problems

Thus, to optimize such a large-scale distribution network, solving about 1000 middle scale (max. 40 cities) TSPs within an interactive length of time (max. 30 seconds) is required. Yet, expert-level accuracy (less than 3% of the deviation from the optimal solution) is always necessary, since domain experts may have such errors in their solutions but never generate worse solutions which may

cause practical problems.

Thus, the authors of this paper developed an efficient method for solving the TSP through elaborating a random restart method. The developed method enables to guarantee the responsiveness through limiting the number of repetitions and through devised component methods and heuristics (Kubota et al. 1999). However, to meet the required guarantee of below 3% of errors, it took more than 1 minute to solve all 1000 TSPs. Thus, the time to solve 1000 TSPs was needed to be decreased.

Therefore, in order to improve the responsiveness, we proposed a GA, where heuristics are applied for the crossover and the mutation as well as its generation number is limited (Onoyama et al. 2000). However, for some kinds of delivery location patterns included in large-scale distribution networks, obtained solutions had more than 3% of errors. Thus, other heuristics were applied to cover the weaknesses of the solving method. However, these heuristics were not effective for some patterns, and the above-mentioned accuracy was still not guaranteed for all kinds of patterns.

In the next section, a knowledgeable approximate method to solve above-mentioned problems is proposed.

## 3. Knowledge-embedded Multi-stage Genetic Algorithm

As stated in earlier sections, the delivery routing problem in the above distribution network simulation can be taken as a TSP, especially a symmetrical (non-directed) Euclidean TSP (Yamamoto and Kubo 1997) assumed in this paper.

### 3.1 Concept of the Proposed Method

In order to solve problems mentioned above in section 2, the following knowledge-embedded multi-stage GA method is proposed to guarantee both responsiveness and accuracy for various kinds of delivery location patterns.

#### (1) Knowledge-embedded multi-stage GA

It is difficult to realize an effective way that always guarantee expert-level optimality for various distribution location patterns with required responsiveness. Heuristics effective to certain patterns are not necessarily useful to others. Yet, application of excessively complicate algorithms or heuristics makes the responsiveness worse. Therefore, a high-speed GA that mainly uses simple general heuristics is combined with a knowledge-embedded GA into which knowledge for handling particular problems is incorporated. In this way, we could avoid local minimum for various delivery location patterns.

Concretely speaking, 2opt-type mutation is used for the high-speed GA. This 2opt-type mutation quickly improves tours. Therefore, good solutions are usually expected to be obtained within a short length of time. However, it also takes risks of falling into local minimum. According to authors' experiments, this high-speed 2opt-type GA brings about inefficient tours for certain delivery location

patterns.

Therefore, knowledge-embedded multi-stage GA method is proposed. In this method, GA (called block-type GA) having the following knowledge to meet with the particularities of problems is applied after the use of the 2opt-type high-speed GA.

Namely, the following rather problem-oriented knowledge about the neighborhood conditions or their relaxation is incorporated into operations of the block-type GA so that these operations can be controlled through utilizing the knowledge. (a) Multi-step NI: Particular heuristics that constructs the initial tour through using NI (Nearest Insertion) method step-by-step to globally consider adjacent delivery locations, where the adjacency is defined by problem-oriented knowledge. (b) Block-type mutation: Select a node randomly out of a tour, and mutate it together with its adjacent nodes in order to avoid local minimum solutions.

#### (2) Limiting the generation number of GA

In this method, time necessary for processing one generation of GA is calculated under the determined factors such as the population size and the probability of crossover and mutation that affect the response time of GA. Then, the number of generations repeatable within the required response time is calculated. Finally, the above calculated number of generations is repeated to obtain the solution as optimal as possible within required response time.

### 3.2 Components of the Proposed Method

#### (1) Method for generating initial individuals

In order to obtain highly optimal solution through avoiding the convergence into local minimum, the randomness of the initial individuals is important. However, the speed of convergence slows down, if totally random initial solutions are generated as is done by the random method a). Thus, the other methods are devised as shown below.

##### a) Random method

Construct a tour through putting nodes in random order.

##### b) Random NI method

Put nodes in random order and using NI method according to the order, reorder the nodes of the tour.

##### c) Multi-step NI method

In case experts generate a traveling route, they usually determine the order of delivery locations, globally considering the whole route, so that the nearest location from the present one can always be the next location to deliver. On the model of such global consideration of experts, a multi-step NI method is proposed which enables to generate a traveling route similar to one generated by experts.

In detail, this method constructs a tour through the following procedure: 1) Temporally adding a node to a tour by way of NI method and let  $A$  be the resultant tour length increased by the temporal addition. 2) Multiply the original tour length ( $= B$ ), before the tour is changed in step 1), by certain weight ( $= w$ ). 3) If  $A < (w * B)$ , the node is inserted (actually added) into a tour. 4) Repeat 1) through 3) until

all nodes satisfying 3) are inserted. 5) For nodes which do not satisfy 3), try 1) through 4) with the weight increased. Here the range of this increase is defined as problem-oriented knowledge.

#### (2) Method for crossover

To inherit good features of parents through crossover and to realize the prompt convergence of solutions in GA, a crossover method using NI method is proposed. This crossover method called NI-combined crossover comprises the following steps: 1) Determine the crossover point in one of parent chromosomes. 2) Obtain a sub-tour represented by a group of genes located before the crossover point in the chromosome. 3) Change the order of remaining nodes that are not contained in the sub-tour obtained in 2), according to the order of node (genes) in the other parent chromosome. 4) Using NI method, insert the remaining nodes into the sub-tour obtained in 2), in the order after reordering in 3).

In this way, the generated tour is represented as a new child. And through applying this NI-combined crossover method, the order of nodes contained in parents is inherited to their children to increase the convergence speed.

#### (3) Method for mutation

Mutation of GA often did not take much effect on the convergence of solutions without combining local search methods or without embedding problem-oriented knowledge. Thus, the following two mutation methods are proposed.

##### a) 2opt-type mutation

This method enables to improve the convergence speed through combining a 2opt-like simple local search heuristic method. This is to say, a gene (representing a node) for mutation, we call it a mutation node, is randomly selected out of parent's genes, and the other node is selected from nodes except the mutation node and its next in the tour. The mutation node and its next node make a link. The other node and its next node make another link. After exchanging these two links, the length of a new tour is evaluated. If the tour length is shorten as a result of the exchange, the trials of such exchange are kept on going through successively changing the other node until the improvement (decrease) in the tour length is found or until such exchanges are all checked.

##### b) Block-type mutation

2opt-type mutation easily improves tours, and good solutions are expected to be obtained within a short length of time. However, it also takes risks of failing into local minimum. To obtain a further optimal solution, it is desirable to escape from local minimum by destroying a part of a tour. For this purpose, the following block-type mutation is proposed.

At first, select a mutation node from a tour at random to delete it together with its neighbor nodes. The size of this neighborhood is also selected at random within the range specified by somewhat problem-oriented knowledge. Then, reconstruct the tour through inserting the deleted nodes into the tour using NI method.

### 3.3 Proposed Solving Method

Through integrating above components, the following three kinds of GA methods are proposed to assure both responsiveness and accuracy for various kinds of delivery location patterns.

#### (1) 2opt-type GA

This uses the random NI method for generating initial individuals, NI-combined crossover for the crossover, and the 2opt-type mutation for the mutation. This method makes it possible to guarantee short time convergence of solutions due to further improving initial solutions generated by random NI method through the application of the NI-combined crossover and the 2opt-type mutation.

#### (2) Block-type GA (Knowledge-embedded GA)

In this method, half of initial individuals are solutions obtained by the multi-step NI method and another half are those of the random method. The NI-combined crossover is used for the crossover operation and the block-type mutation is used for the mutation. This method is considered to obtain highly optimal solutions through avoiding local minimum due to constructing the highly random initial solutions mixed with globally near optimized ones and due to reconstructing a large part of a locally optimized tour by the use of block-type mutation.

#### (3) Knowledge-embedded multi-stage GA

The finally proposed method is called "Knowledge embedded multi-stage GA". This comprises the 2opt type GA method followed by the block type GA method. And the knowledge-embedded multi-stage GA selects the better one out of the solutions obtained by use of these two GA methods in order to have highly accurate solutions for coping with various types of delivery location patterns. Yet, to guarantee the responsiveness, both of these two GAs finish processing within the limited length of time through initially calculating the number of generations repeatable within the time limit (e.g. 15 milliseconds for each GA).

### 4. Experiment and Result

#### 4.1 Experiment

In this section, the experiment to evaluate the proposed method is explained. A computer equipped with Intel Pentium II (450MHz) processor and 256MB memory is used for this experiment. In distribution networks targeted, a truck cannot go around more than 40 delivery locations (parts makers) within one day. Therefore, 40 cities TSPs were used for this experiment. Yet, various combinations of 40 delivery locations are possible. Thus, randomly selected 20000 different patterns of 40 delivery locations were prepared. Then, to evaluate three kinds of GA methods described in 3.3, each solving method solved 20000 test patterns for 100 times and the probability to obtain solutions within 3% of errors was calculated.

#### 4.2 Result

To guarantee the responsiveness, the time necessary for processing one generation is calculated, and based on this value, the generation number of GA is determined. Table 1

shows an example of the generation number to respond within 30 milliseconds when the population size is 100. Then, the tests were repeated 100 times for three kinds of GAs. Each test used 20000 kinds of delivery location patterns. The probability to obtain solutions within 3% of errors compared to the optimal solutions was checked. Furthermore, the probability to obtain the optimal solutions within 30 milliseconds was also checked. These results are shown in Table 2.

**Table 1. The number of generations of each method repeatable within 30 milliseconds**

#	Method	Initial Generations	Mutation	Generations
1	2opt-type GA	Random NI	2opt-type	24
2	Block-type GA	Random + Multi-step NI	Block-type	20

**Table 2. The solution optimality**

#	Method	Optimal (%)	Under 3% error (%)
1	2opt-type GA	84.45	99.885
2	Block-type GA	83.75	99.785
3	Multi-stage GA	92.05	100.0

### 5. Evaluation

According to Table 2, only the knowledge-embedded multi-stage GA method could solve a 40 cities TSP with less than 3% of errors with 100% of probability within 30 milliseconds.

#### (1) Effect of block-type GA (knowledge-embedded GA)

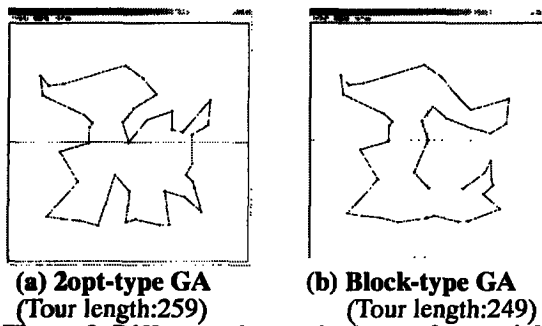
Tour's shapes were examined as to solutions generated by the 2opt-type GA and leaving more than 3% errors. As a result, most of these shapes were like gear wheels as shown in fig. 3 (a). Experts usually generate more straight routes as shown in fig. 3 (b). If experts find inefficient routes such as shown in fig. 3 (a), they reject to use the system since they consider it as unreliable one. In case of using block-type GA (knowledge-embedded GA), tours similar to fig. 3(b) were generated even for such delivery location patterns. The reason is that knowledge-embedded GA integrates the random method, the multi-step NI, and the block-type mutation in order to avoid falling local minimum.

#### (2) Effect of knowledge-embedded multi-stage GA

According to our experiment, in case of the 2opt-type GA, 23 cases out of 20000 tests had over 3% errors. In case of the block-type GA method, 43 cases had more than 3% errors.

However, the knowledge-embedded multi-stage GA, namely, the 2opt-type GA subsequently followed by the block-type GA could generate solutions below 3% of error within 30 milliseconds, for every case in 20000 tests. The reason is that, coping with various delivery location patterns, either 2opt-type GA or knowledge-embedded GA can avoid falling into local minimum (over 3% errors). Thus, the knowledge-embedded multi-stage GA method could guarantee the responsiveness as well as the expert-level accuracy, namely, below 3% errors.

expert-level accuracy, namely, below 3% errors.



**Figure 3. Difference in tour's shape of a special location pattern**

## 6. Comparisons

A lot of methods to solve TSP are proposed for practical applications. In this section, our methods are compared with other methods.

LK and its improving methods (Lin and Kernighan 1972; Yamamoto and Kubo 1997) take a long calculation time. For example, it took 40 seconds for the LK method to solve a 40 cities TSP. This long calculation time makes it unsuitable to apply these methods for interactive distribution logistic simulations.

Theoretically, SA (Yamamoto and Kubo 1997) is said to be able to search very near-optimal solutions by decreasing the risk of falling into local minimum. But practically, it is very difficult to adjust SA's parameters such as cooling speed for coping with various location patterns. Furthermore, SA usually takes a long calculation time to get above-mentioned theoretical near-optimal solutions. Also, Tabu Search (Hooker and Natraj 1995) usually needs a long calculation time to get practically optimal solutions. In some of our experiments, it took about 400 milliseconds for SA to solve a TSP of 15 cities and about 40 seconds to solve a TSP of 100 cities. Therefore, these methods are not suitable for repetitive simulations including interactive human judgements such as our application.

Moreover, some algorithms that can search very near-optimal solutions for the Euclidean TSP in polynomial time using devised DP are proposed (Arora 1998). However, these algorithms also take too long time to use for practical applications such as ours and it seems too hard for ordinary system developers to modify them flexibly for coping with various special requirements of practical applications.

So-called random restart methods which apply local search such as 2opt for improving random initial solutions, can obtain near-optimal solutions. These include GRASP (Feo, Recende and Smith 1994) or the elaborated random restart method (Kubota et al. 1999) that can guarantee responsiveness by limiting the number of repetitions. However, according to the authors' experiment, the above-mentioned elaborated random restart method needed

about 100 milliseconds to solve the 40 cities TSP and to guarantee less than 3% errors (Kubota et al. 1999).

## 7. Conclusion

In this paper, a knowledgeable GA method for solving the TSP was proposed and evaluated. This is applicable to the optimization of large-scale distribution networks that requires repetitive interactive simulations. This kind of application requires responsiveness as well as optimality, for example, solving 1000 TSPs with expert-level accuracy within 30 seconds.

In order to guarantee expert-level solutions for various kinds of delivery location patterns, the high-speed GA was combined with the knowledge-embedded GA. The high-speed GA comprises the random NI method and the 2opt-type mutation. And this high-speed GA mainly uses simple general heuristics. The knowledge-embedded GA includes the random method, the multi-step NI method, and the block-type mutation. And particular knowledge was incorporated in this knowledge-embedded GA to make up for the weakness of the high-speed GA. Namely, to cope with delivery location patterns for which the high-speed GA cannot guarantee expert-level solutions, this knowledge-embedded GA has rather problem-oriented knowledge.

According to our experiment, in case of using the former high-speed GA, 23 test cases out of 20000 test cases had more than 3% of errors compared to the optimal solution. However, our proposed knowledge-embedded multi-stage GA method (which comprises the high-speed GA and the knowledge-embedded GA) could solve each of all 20000 test cases within 30 milliseconds at expert-level accuracy (less than 3% errors).

## References

- Arora, S. 1998. Polynomial Time Approximation Schemes for Euclidean TSP and Other Geometric Problems, *Journal of the ACM*, Vol. 45, No.5, pp.753-782.
- Feo T.A.; Recende M. G.C.; and Smith S. H. 1994. A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set, *Operations Research*, Vol. 42, No.5, pp.860-878.
- Hooker, J.H.; and Natraj, N.R. 1995. Solving a General Routing and Scheduling Problem by Chain Decomposition and Tabu Search, *Transportation Science*, Vol.29, No.1, pp.30-44.
- Ibaraki, T. 1993. *Discrete Optimization and Algorithm*, Iwanami Syoten, Tokyo.
- Kubota S.; Onoyama T.; Onayagi K. and Tsuruta S. 1999. Traveling Salesman Problem Solving Method fit for Interactive Repetitive Simulation of Large-scale Distribution Network, *Proc. IEEE SMC'99*.
- Lin, S.; and Kernighan, B.W. 1972. An effective heuristic algorithm for the traveling salesman problem, *Operations Research*, Vol. 21, No.2, pp.498-516.
- Onoyama, T.; Kubota, S.; Oyanagi, K.; and Tsuruta, S.: A Method for Solving Nested Combinatorial Optimization Problems , *Proc. IEEE SMC2000*.
- Yamamoto, Y.; and Kubo, M. 1997. *Invitation to Traveling Salesman Problem*, Asakura Syoten, Tokyo.