Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Multi-inner-world Genetic Algorithm using Multiple Heuristics to Optimize Delivery Schedule

Yoshitaka Sakurai, Setsuo Tsuruta
School of Information Environment
Tokyo Denki University
Chiba, Japan
{ysakurai, tsuruta}@sie.dendai.ac.jp

Takashi Onoyama, Sen Kubota
Research & Development Department
Hitachi Software Engineering Co.,Ltd.
Tokyo, Japan
{onoyama, kubosen}@hitachisoft.jp

***Abstract*— Building a delivery route optimization system that improves the delivery efficiency in real time requires to solve several tens to hundreds cities Traveling Salesman Problems (TSP) within interactive response time, with expert-level accuracy (less than 3% of errors). To meet these requirements, a multi-inner-world Genetic Algorithm (Miw-GA) method is developed. This method combines several types of GA's inner worlds. Each world of this method uses a different type of heuristics such as a 2-opt type mutation world and a block (Nearest Insertion) type mutation world. Comparison based on the results of 1000 times experiments proved the method is superior to others.**

***Keywords*— Traveling Salesman Problems (TSP), Genetic Algorithm (GA), Heuristics**

I. INTRODUCTION

Due to the complicated road network, the efficiency of product distribution remains on a lower level in Japan compared to that of the USA, which disadvantages the productivity of Japanese industries. This inefficiency also causes social problems and economical losses. Namely, we are facing the necessity of urgently reducing the volume of car exhaust gases to meet environmental requirement as well as curtailing transport expenses in Japan.

There are many distribution systems that should be optimized, including the delivery of parcels, letters and products supply/distribution across multiple enterprises. In order to improve the efficiency of these distributions, it is necessary to optimize the delivery routes, or the delivery order of multiple delivery locations (addresses). One round delivery comprises more than several tens or hundreds of different locations. Thus, the optimization of a delivery route can be modeled as such a large-scale of Traveling Salesman Problem (TSP). However, TSP is a combinatorial problem that causes computational explosion due to n! order of combinations for n-city TSP. Therefore, to practically obtain the efficient delivery route of such a distribution system, a near optimal solving method of TSP is indispensable. Yet, the practical use of such a solving method on an actual site needs human confirmation (which is difficult to formulate) of the solution, since social and human conditions are involved. Namely, human users should check to understand that the solution is practical. Users sometimes should correct manually or select the alternative solution.

Therefore, the TSP solving methods are required to ensure the response time necessary for the above human interaction.

By the way, 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 TSP solving methods [1] [2] may generate even mathematically optimal solutions in some cases but cannot ensure the amount of errors below 2~3%. Such errors possibly discourage user, which makes those conventional methods not practically useful, especially for the above-mentioned applications.

Thus, we have already proposed several types of genetic algorithm (GA) using simple heuristics to interactively optimize TSP. These GA had good performance at a certain problem pattern, but has tendency to fall into local minimum solution at another pattern. To cope with, we found that block type GA using nearest insertion (NI) can compensate 2-opt type GA. Thus, through cascading these two types of GA, we proposed a Multi-outer-world Genetic Algorithum (Mow-GA). However, this has limitation to increase efficiency or optimality [3].

This paper proposes a Multi-inner-world Genetic Algorithm (Miw-GA) using multiple heuristics in each generation of one GA. "inner" means this. This can efficiently solves the TSP with less dependency of such problem pattern. This method enables to guarantee the responsiveness by limiting the number of generations of GA and by integrating several kinds of genetic operators (initial generations, mutation, and crossover) each having selected heuristics and its parameters. Fundamentally, the proposed Miw-GA method combines two types of inner worlds, one of which uses a 2-opt type mutation and another uses a block (nearest insertion) type mutation.

The paper is organized as follows: In the next (second) section, the delivery route optimization problem 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 experiments, and in the sixth section, we will compare it with other methods. And in the last seventh section, the results will be concluded.

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

II. PROBLEMS IN DELIVERY ROUTE OPTIMIZATION

In this section, firstly, two kinds of actual distribution systems are depicted. And, the optimization problems of these distribution systems are formally and technically described.

A. Delivery Route Optimization Problem

In order to optimize the above-mentioned large-scale distribution network, we need to grasp the total cost of distribution under various conditions through repeating the simulation process. To globally evaluate these results, human judgment is indispensable and interactive response time is required.

At the delivery route optimization problem for parcels and letters, a round delivery is carried out 1-3 times a day with a small vehicle such as a motorcycle or a small truck. Delivery zone that is covered by one vehicle is different according to the region. Delivery locations are comparatively overcrowded in the urban area, whereas scattered in the rural area. Therefore, the number of locations (addresses) for delivery differs - over several tens or hundreds - depending on the region and time zone. It is necessary to make and optimize a new delivery route for each round delivery since delivery locations change every day and every time. Though human or social factors should be considered, this is a problem to search the shortest path or route, modeled as a famous “Chinese Postman Problem” or “Traveling Salesman Problem (TSP)”. The computer support by near optimal solving method is quite useful to reduce the burden and loss time of workers as well as car exhaust gases in such distribution networks or parcels /letters delivery.

B. Problem Specification

The delivery route optimization problem of these distribution systems is formulated as follows:

The delivery network is represented by weighted complete graph G=(V,E,w). V is node set. A node v_i （i=1,…,N） represents a location (address) for delivery. N is the number of nodes. E is edge set. A edge e_{ij} represents a route from v_i to v_j. w is edge weight set. A edge weight d_{ij} represents a distance from node v_i to node v_j, $d_{ij} = d_{ji}$. The problem to find the minimal-length Hamilton path in such a graph G=(V,E,w) is called Traveling Salesman Problem (TSP).

Thus, to improve the delivery efficiency of such distribution systems, it is required to obtain an approximate solution of a TSP within an interactive length of time (max. hundreds of milliseconds). 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.

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

III. A MULTI-INNER-WORLD GENETIC ALGORITHM

As stated in the foregoing sections, the delivery routing problem in the above distribution systems can be formalized as a TSP. Especially a symmetrical (non-directed) Euclidean TSP [1], [2] is assumed in this paper.

A. Concept of the Proposed Method

In this paper, Multi-inner-world GA (Miw-GA) is proposed. In this method, one GA has several kinds of GA worlds. This guarantees both real-time responsiveness and accuracy for various kinds of delivery location patterns. At the initial phase of GA, groups of individuals (population) that become the candidates of the solution are generated. And, based on the population, new individuals (solution candidates) are generated by the mutation operator, and individuals are improved by the evaluation and the selection. With our GA, each individual (chromosome) represents the tour, namely the delivery route in TSP. Each gene of the chromosome represents the node number (identification number of the address for delivery). A chromosome is a sequence of nodes whose alignment represents a round order as shown in Fig.1.

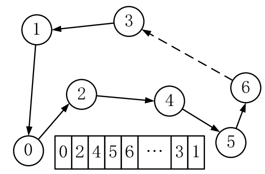

Fig. 1. Chromosome.

1) Multi-inner-world GA: It is difficult to find an effective search method that always guarantees expert-level optimality as well as the required real-time behavior for various distribution location patterns. Heuristics, that are effective to particular patterns, are not necessarily useful to other patterns. Yet, the application of excessively complicate algorithms or heuristics makes the responsiveness worse. Therefore, genetic operators that implemented several kinds of heuristics are used for improving solutions.

In this paper, as for genetic operators, a 2-opt-type mutation and block (NI) type mutation are used. This 2-opt-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 a local minimum. Our experiments revealed that 2-opt-type mutation implemented GA (called 2-opt-type GA) computes some inefficient tours for certain delivery location patterns. On the other hand, as for block-type mutation, the optimization ability is high and the computational cost is also relatively high. Moreover, block-type mutation can efficiently solve problems that 2-opt-type mutation subject to fall into local minimum. In this way, Multi-inner-world GA (Miw-GA) could avoid local minima for various delivery location patterns by using several kinds of heuristics.

B. Components of the Proposed Method

GA operators are composed of initial individual generation, individual improvement (mutation, crossover), and individual selection. Generally, crossover operators are effective for large-scale problems but it needs high calculation cost. In the proposed GA, crossover operators are not used because the objective is to obtain approximate solutions of tens to hundreds cities TSP within interactive response time.

1) Method for generating initial individuals: In order to obtain a highly optimized solution by avoiding the convergence into a 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 a random method. Thus, the other method called Radom NI method is devised as shown below.

Random NI method puts nodes in a random order and inserts them into a tour, using a NI (Nearest Insertion) method in randomized order.

A pseudo code of NI method that inserts a node x_{new} to sub-tour$\{x_1, x_2, ..., x_n\}$ is described as follows:

begin

L_{before} := length of sub-tour $\{x_1, x_{new}, x_2, ..., x_n\}$;

for (j = 2, 3, … , n) do

remove node x_{new} from sub-tour;

insert node x_{new} between x_j and x_{j+1} of sub-tour;

L_{after} := length of sub-tour $\{x_1, \ldots, x_j, x_{new}, x_{j+1}, \ldots, x_n\}$;

if ($L_{after} < L_{before}$)

then $L_{before} := L_{after}$; Insersion location i = j;

insert node x_{new} between x_i and x_{i+1} of sub-tour;

end

2) Method for mutation: Mutation of GAs often did not have much impact 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) 2-opt-type mutation: This method enables to improve the convergence speed by combining a 2-opt-like simple local search heuristic method with GA's mutation operation. This consists of the following steps:

(1) $tour_{par} = \{x_1, x_2, ..., x_n\}$ is a parent tour and $tour_{chi}$ is a child tour.
(2) Copy the contents of $tour_{par}$ to $tour_{chi}$.
(3) Select a node x_i randomly from $tour_{chi}$.
(4) Select another node x_j randomly from $tour_{chi}$ except $\{x_i, x_{i+1}\}$.
(5) Generate $tour_{gen}$ $\{x_1, \ldots, x_i, x_j, \ldots, x_{i+1}, x_{j+1}, \ldots, x_n\}$ by reversing sub-tour $\{x_{i+1}, \ldots, x_j\}$ of $tour_{chi}$ $\{x_1, \ldots, x_i, x_{i+1}, \ldots, x_j, x_{j+1}, \ldots, x_n\}$
(6) If $L_{chi} < L_{gen}$ (tour length is not improved), then it ends. Else copy the contents of $tour_{gen}$ to $tour_{chi}$. Until such link exchanges are all checked, return to step (4) and repeat. L_{gen} is the length of $tour_{gen}$. L_{chi} is the length of $tour_{chi}$.

b) Block-type mutation: 2-opt-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 a local minimum. To obtain a solution closer to the optimum, it is desirable to escape from a local minimum by destroying a block of a tour at a time. For this purpose, the following block-type mutation is proposed. This consists of the following steps:

(1) $tour_{par} = \{x_1, x_2, ..., x_n\}$ is a parent tour. $tour_{chi}$ is a child tour.
(2) Select a node x_i randomly from $tour_{par}$.
(3) Move the nodes, except $\{x_{i-r}, \ldots, x_{i+r}\}$ namely except x_i and its neighbor nodes of $tour_{par}$, to $tour_{chi}$. The size of neighborhood r is specified as problem-oriented knowledge, for instance, a random number from *0* to *B * (the distance to the node farthest from a depot)*. *B* is a constant number specified as problem-oriented knowledge.
(4) Insert $\{x_{i-r}, \ldots, x_{i+r}\}$ into $tour_{chi}$ using the NI method. When all nodes have been inserted to $tour_{chi}$, the mutation processing ends.

3) Method for selection: In order to get highly optimized solutions and realize quick convergence in GAs, individuals are selected out of the population including both parents' and children's. And, 10% of individuals in a new generation are selected randomly from the above populations to give the chance of reproduction to even inferior individuals. Furthermore, to enhance the evolution efficiency, only one individual is selected when the same individuals are generated.

C. Proposed Solving Method

Through integrating above components, Multi-inner-world GA is proposed to ensure both real-time responsiveness and accuracy for various kinds of delivery location patterns.

This method is shown in Fig.2. This method makes it possible to guarantee quick convergence of solutions through improving initial solutions due to the random NI method and through applying the block-type mutation and the 2-opt-type mutation.

1) Multi-inner-world GA: The proposed method is called "Multi-inner-world GA (Miw-GA)". This has a 2-opt-type mutation world followed by a block type mutation world. Miw-GA selects the one out of these two mutation worlds. As for its selection (scheduling), there are random (probabilistically changing opportunity) scheduling, round robin (equal opportunity) scheduling and so on. This raises the probability to have highly accurate solutions for various types of delivery location patterns within an interactive real-time context, because of the following reasons:

Though the computation time of the block (NI) type mutation is O(n2), the computation time of the 2-opt-type mutation is much smaller than the former. Furthermore, the NI method checks just links among neighbors but all links among neighbors in the tour to be inserted. Meanwhile, though not all links, the 2-opt operation in the 2-opt-type mutation checks links between nodes that are not neighbors. Thus the 2-opt-type mutation but not being in the block type mutation can have the possibility to search other optimal solutions than the NI method, namely the block type mutation where only NI method is used effectively as heuristics.

Yet, to guarantee real-time responsiveness, the GA finishes their processing within the limited length of time due to the offline calculation result concerning the number of generations repeatable within the time limit.

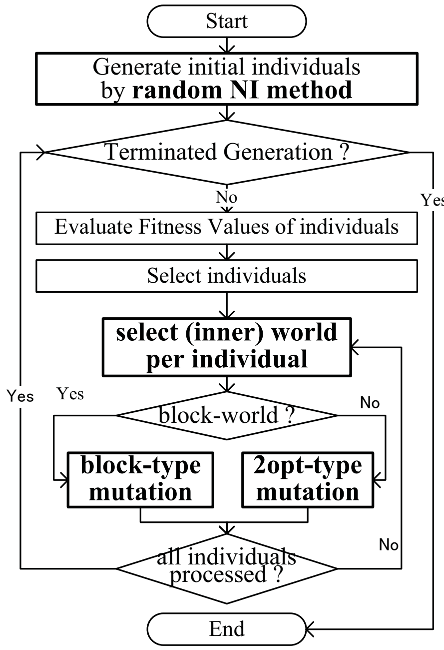

Fig. 2. Miw-GA

IV. Related Works

As for the approximate solution technique, various techniques are proposed. Lin-Kernighan (LK) method is famous as the heuristics search technique for TSP. However, LK and its improving methods [4], [2] also take a long computation time though the optimality of obtained solutions is high and these methods are often incorporated with the meta-heuristics search such as SA, TA and GA.

Simulated Annealing (SA) and Tabu Search (TS) are known as the meta-heuristics search technique. Theoretically, SA [5] is said to be able to search very near-optimal solutions by decreasing the risk of falling into a 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 computation time to get above-mentioned theoretical near-optimal solutions.

TS [6] usually needs a long computation time to get practically optimal solutions. Worse still, TS is said to be weak in maintaining solution diversity though it has strong capability for local search. However, TS is improved in these weaknesses by Kanazawa et al. [7].

So-called random restart methods [8], which apply local search such as 2-opt for improving random initial solutions, can obtain near-optimal solutions. These include Greedy Randomized Adaptive Search Procedure (GRASP) [9] or the elaborated random restart method [10] that can guarantee responsiveness by limiting the number of repetitions. However, according to our experiments, even the above-mentioned elaborated random restart method needed about 100 milliseconds to solve 40 cities TSP and to guarantee less than 3% errors [10].

As for the Genetic Algorithms (GA) to efficiently solve TSP, various techniques are proposed. GA applied solving methods using the edges assembly crossover (EAX) [11] and the distance-preserving crossover (DPX) [12] could get highly optimized solutions in case of very-large-scale TSPs (with 1000-10000 cities) [13], [14]. These crossover methods examine characteristics of parent's tour edge to strictly inherit to children. However, since these crossover operations take long computation time for analyzing edges, using it for not-very -large-scale TSP is often inefficient.

In reference [13], two kinds of methods are compared in many cases. It shows that Cga-LK is advantageous to 300-10000 cities TSP, but Random-LK is advantageous to 198 cities TSP. Therefore, the solution that can efficiently solve TSP of 1000 cities or more can not necessarily efficiently solve TSP of about 100 cities. As to TSP of our intended scale (with 10s to 100 cities), in reference [13], a TSP lin105 is solved with 1.77% average error rate in 231seconds.

Moreover, in reference [15], the performance comparison experiments were conducted using various crossover operators.

A GA method with the same purpose as ours (aiming to obtain high quality approximate solution as fast as possible for 10s - 100s cities TSPs) is proposed by Yan et al [16].

In next section, Random-LK in reference [13] and the best crossover operator of experiments in reference [15] and TS by Kanazawa and GA by Yan are compared with proposed method.

V. Experiment and Results

A. Experiment

In this section, to compare the proposed method with other methods, the comparison experiment is done by solving the TSPLIB problem.

Experiments were conducted under the following computation environment. Namely, CPU is AMD Athlon 64 X2 3800+ 2GHz processor. It is almost the same performance as Athlon 64 3200+ 2GHz because of its execution on the single core mode with 1GB memory. The programs were written in C language, compiled by Microsoft Visual C++ .NET 2003 ver. 7.1.3091 with /O2 option (directing the execution speed preference), and executed on Windows XP Professional.

Yet, since other solution methods to be compared are executed on machines with different performance specification, it is necessary to take the difference into account. Therefore, referring to the statistical results of tests using RC5-72 benchmark [17] for measuring the arithmetic processing speed, we obtained the spec difference correction coefficient (SDCC). This can be obtained by dividing the resultant value of the benchmark test executed on our experimental environments, by the resultant value of the benchmark test on the experimental environment of other solution methods. Through multiplying

SDCC to the execution time of other solution methods, we calculated an assumed execution time on the same specification machine as ours.

To compare the proposed methods with Random-LK in reference [13] and the best crossover operator of experiments in reference [15], the proposed methods are tested on lin105 TSP in TSPLIB.

To compare the proposed methods with GA by Yan and TS by Kanazawa, the proposed methods are tested on nine benchmark problems in TSPLIB whose number of cities ranges from 70 to 280. As to GA parameters, mutation rate is 40%, chosen probability of mutation type is block-type 60%, 2-opt-type 40%. These parameters are settled based on pre-experiment results. Each problem is solved 1000 times. Miw-GA stops searching when the computing time exceeds average computing time of the method to be compared.

B. Results

Table 1 presents the SDCC of each method.

Random-LK in reference [13] solved lin105 with 1.77% average error rates in 231seconds on 200-MHz PentiumPro PC running Linux 2.2.12. Since this SDCC is 0.048, the solving time on our experimental environment is 11.088 seconds.

The best crossover operator of experiments in the reference [15] solved lin105 with 3.1% average error rate in 750 seconds on SUN SPARC Ultra-5 10 machine. Since this SDCC is 0.065, the solving time on our environment is 48.75 seconds.

Meanwhile, our Miw-GA obtains the optimal solution within 1.11 seconds, and obtains a solution with average error rate 0.31% in 0.15 seconds. Miw-GA is far superior to the above two techniques in speed and in optimality for lin105.

Table 2 presents the experimental results obtained by applying Miw-GA to the above nine benchmark problems and results corrected by using SDCC. The mark “-” on the Table 5 indicates no data. The digits (e.g. 70) contained in the name (e.g. st70) of TSP indicate the number of cities.

TABLE I
Spec Difference Correction Coefficient (SDCC)

	Spec	SDCC
Random-LK	CPU:PentiumPro 200-MHz, OS:Linux 2.2.12.	0.048
in reference (Cheng 2002)	SUN SPARC Ultra-5 10 machine	0.065
TS by Kanazawa	CPU:Pentium4 2.55GHz, memory:1GB (DDR266)	0.590
GA by Yan	CPU:Pentium4 2.4GHz, memory:256MB	0.519

Results of GA by Yan are compared with those of Miw-GA. All the results excluding pr144 indicate Miw-GA can obtain the solution superior to GA by Yan. Results indicate Miw-GA can obtain solution accuracy superior to GA by Yan. Furthermore, the computation time is much less such as 10% for st70, 20% for eil76, 14% for kroA100, 2% computation time for a280. Specific results of problem a280 indicate Miw-GA can obtain solutions whose average error rate is 0.722% which is lower than that of GA by Yan (7%) at the same computation time.

Next, results of TS by Kanazawa and Miw-GA are compared. All the results indicate Miw-GA can obtain the solution superior to TS by Kanazawa. Results indicate Miw-GA obtained superior accuracy solutions to TS by Kanazawa, while the computation time is 10% for the problem pr107, 93% for pr144, 8% for pr152, 3% for pr226.

Overall results show that Miw-GA is much superior to GA by Yan and TS by Kanazawa in solving the above mentioned nine TSP benchmark problems whose number of cities ranges from 70 to 280.

TABLE II
Results compared with related works on TSPLIB

Name of TSP	Average error rate from optimal solution [%] (Average execution time [sec])			
	GA by Yan	TS by Kanazawa	Miw-GA same time	Miw-GA same error
st70	0.312 (0.348)	-	0.168 (0.359)	0.276 (0.034)
eil76	1.184 (0.602)	-	0.769 (0.609)	1.173 (0.123)
kroA100	0.016 (0.877)	-	0.000 (0.890)	0.014 (0.123)
pr107	-	0.290 (0.826)	0.008 (0.828)	0.243 (0.079)
pr144	0 (4.136)	0.019 (4.685)	0 (4.390)	0 (4.390)
pr152	-	0.120 (7.558)	0.098 (7.765)	0.120 (0.636)
pr226	-	0.510 (12.685)	0 (12.690)	0.447 (0.420)
a280	10.770 (17.371)	-	0.722 (17.381)	7.244 (0.290)

*. Under lined results are best of 3 methods.

C. Analysis

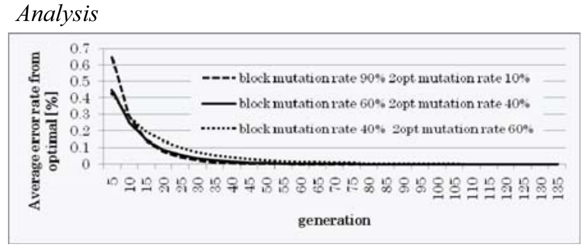

Fig. 3. Learning curve of Miw-GA at kroA100

Fig. 3 shows a learning curve of Miw-GA at kroA100. This results indicates the case of low block-type mutation rate (40%, 60%) is superior to the case of high block-type mutation rate (90%) during early (1st-10th) generation. During middle (11th-100th) generation, the case of high block-type mutation rate (90%) is best. However, at final (after 100th) generation, the case of middle block-type mutation rate (60%) is best. This tendency was observed in all problems except pr152. Optimal mutation rate of block-type versus 2-opt-type was different depending on the problem. Generally, a good performance was observed when block-type mutation rate is around 60%-90%.

Next, the effect of dynamically changing the mutation type selection method was experimentally analyzed. The experiment was done for two cases in the same condition as the above comparison experiment. In Case 1, one out of the above two mutation types was selected always at probability of 50%. In Case 2, mutation types were selected based on individual selection history. At first, block-type mutation was selected for all individuals. After that, the other mutation type than the one selected just before was selected.

As a result, Case 2 obtained better results for 4 problems (st70, eil76, pr152, a280) in 8 questions (same as the above comparison experiment). Other cases showed almost the same performance.

VI. CONCLUSION

In this paper, Multi-inner-world GA method for solving the TSP was proposed and evaluated. This is applicable to the optimization of various distribution systems such as the parcel and letter delivery as well as large-scale distribution networks that requires repetitive interactive simulations. This kind of application requires responsiveness as well as optimality, for example, solving a TSP with expert-level accuracy within tens or hundreds of milliseconds.

Our experimental results showed that the proposed methods enable to solve TSPs with above-mentioned responsiveness and optimality. These results also showed that performance (computational cost, optimality) of the method is superior to other related works.

Analysis results of the learning curve of Miw-GA clarified the effectiveness of the optimizing method through dynamically changing the block-type mutation rate and 2-opt-type mutation rate depending on the generation.

High performance in optimizing delivery routes was shown as the effect of integrating two heuristics (2-opt and block) into GA. The research related to other heuristics remains for the future.

ACKNOWLEDGMENT

This work was partially supported by Research Institute for Science and Technology of Tokyo Denki University (Q08J-07) and the Kayamori Foundation of Informational Science Advancement.

REFERENCES

[1] Lawer, E., Lenstra, J., Rinnony, K., and Shmoys, D., “The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization”, John Wiley Sons, 1985.

[2] Yamamoto, Y., and Kubo, M., “Invitation to Traveling Salesman Problem”, Asakura Syoten, Tokyo, 1997.

[3] Y. Sakurai, T. Onoyama, S. Kubota, Y. Nakamura, S. Tsuruta: "A Multi-world Intelligent Genetic Algorithm to Interactively Optimize Large-scale TSP", Proc. of The 2006 IEEE International Conference on Information Reuse and Integration (IEEE IRI2006), Hawaii, USA, pp248-255, 2006.

[4] Lin, S. and Kernighan, B.W., “An effective heuristic algorithm for the traveling salesman problem”, Operations Research, Vol. 21, No.2, pp.498-516, 1972.

[5] Miki, M., Hiroyasu, T, Jitta, T., “Adaptive Simulated Anealing for maximum temperature”, Systems, Man and Cyernetics, 2003. IEEE International Conference, vol.1, pp.0-25, 2003.

[6] Fang, Y., Liu, G., He, Y., Qiu, Y., “Tabu search algorithm based on insertion method”, Proceedings of the 2003 International Conference on Neural Networks and Signal Processing, pp.420-423, 2003.

[7] Kanazawa, T., Yasuda, K., “Proximate Optimality Principle Based Tabu Search”, IEEJ Transactions on Electronics, Information and Systems, Vol.124, No.3, pp.912-920, 2004.

[8] Yanagiura, M. and Ibaraki, T., “On Metaheuristic Algorithms for Combinatorial Optimization Problems”, Transactions of the IEICE, Vol.J85-D-II, No.8, pp.3-25, 2000.

[9] Feo, T.A., Recende, M.G.C., and Smith S. H., “A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set”, Operations Research, Vol. 42, No.5, pp.860-878, 1994.

[10] Kubota, S., Onoyama, T., Onayagi, K. and Tsuruta, S., “Traveling Salesman Problem Solving Method fit for Interactive Repetitive Simulation of Large-scale Distribution Network”, Proc. IEEE SMC’99, pp.533-538, 1999.

[11] Nagata, Y. and Kobayashi, S., “The Proposal and Evaluation of a Crossover for Traveling Salesman Problems: Edge Assembly Crossover”, Journal of Japan Society for AI, Vol.14, No.5, pp.848-859, 1999.

[12] Whiteley, D. and Starkweather, T., “Scheduling Problem and Traveling Salesman: The Genetic Edge Recombination Operator”, Proceeding of ICGA'89, pp.133-140, 1989.

[13] Baraglia, R., Hidalgo, J.I., Perego, R., “A hybrid heuristic for the traveling salesman problem”, IEEE Transactions on Evolutionary Computation, Vol.5, Issue 6, pp.613-622, 2001.

[14] Nguyen, H.D., Yoshihara, I., Yamamori, K., Yasunaga, M., “Implementation of an Effective Hybrid GA for Large-Scale Traveling Salesman Problems”, IEEE Transactions on Systems, Man and Cybernetics, Part.B, Vol.37, Issue. 1, pp.92-99, 2007.

[15] Cheng, C., Lee, W., Wong, K., “A genetic algorithm-based clustering approach for database partitioning”, IEEE Transactions on Systems, Man and Cybernetics, Part C, Vol.32, Issue 3, pp.215-230, 2002.

[16] Yan, X., Liu, H., Yan, J., Wu, Q., “A Fast Evolutionary Algorithm for Traveling Salesman Problem”, Proc. of Third International Conference on Natural Computation (ICNC 2007), Vol.4, pp.85-90, 2007.

[17] JLUG, “Benchmark - JLUG RC5-72 Cracking”, http://www.orange.co.jp/~masaki/rc572/, 2008.