A-034

Application of Genetic Recombination to Genetic Local Search in TSP

PENG gang[1] Ichiro IIMURA[2] Takeshi NAKATSURU[3] Shigeru NAKAYAMA[4]

Abstract In this paper, an approach based upon Genetic Recombination is proposed, and applied to the TSP. The algorithm is composed of two SGAs which only consist of the basic genetic operators such as selection, crossover and mutation. One of the SGAs is named as the Global Genetic Algorithm (GGA), and encompasses the main tours where it is designed to search for the global optimal solutions. Another one is named as the Local Genetic Algorithm (LGA) and traverses over the sub tours to find the local optimal solutions. The LGA is combined to the GGA as an operator. The local optimal solutions are recombined to the main tours for improving the search quality. To investigate the features of the proposed algorithm, it is applied to a small double circles TSP, and some interesting results are presented in our experiments.

1. Introduction

TSP is one of the well-studied combinatorial optimization problems. Many researchers from various fields have devoted to developing new algorithms for solving it. It has been proved that the hybrid of different algorithms is more effective than a single algorithm. For example, the local search has been successful for improving GAs in the search processes [1],[9],[10]. Most of the works on solving the TSP are focused on the efficiency of how to solve the larger TSP instances.
Some of the works aim at expanding the theories of search algorithms, especially in the GAs domain.

In this paper, our attempt is to make a discussion on the GAs based on Genetic Recombination. The algorithm is composed of two SGAs which only contain basic genetic operators. One is the GGA which is applied to the main tours, for searching for the global optimal solutions. Another is the LGA which is applied to the sub tours for searching for local optimal solutions. The SGA was developed by John Holland, and his original algorithm is approximately the same as shown in the Fig.1. [5]. In the early studies, the SGA played an important role in the development of GAs, and attracted the researchers' attentions widely. Goldberg made a detailed discussion on how the SGA works with some simple optimal mathematical functions and other problems in his book [3]. Reeves discussed the differences, and similarities between the SGA and the neighborhood search [2]. Computer scientist Michael D. Vose provided an introduction to what is known about SGA theory. He also made available algorithms for the computation of mathematical objects related to SGA [7]. All the studies have shown that the SGA is still valuable in heuristic search algorithms.

1,3. Oita National College of Technology
2. Department of Administration, Faculty of Administration, Prefectural University of Kumamoto
4. Faculty of Engineering, Kagoshima University

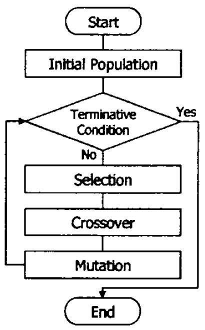


Fig.1. Simple GA

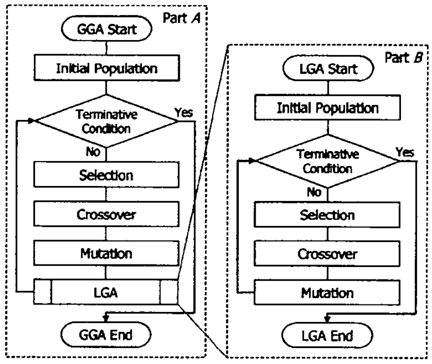


Fig.2. New Algorithm

Our approach is to use a sub tour which contains a set of continuous cities ordered according to the visiting order. The basic idea is to find a better sub tour to replace the original one. This operation acts like the Genetic Recombination in the Genetic Engineering field. Genetic Recombination is the process by which the combination of genes in an organism's offspring are different from the combination of genes in that organism. In the TSP, it is the technique related to the local search. To find the better cities in the tour, we chose a set of contiguous cities from the main tour to form a sub tour, and then apply the LGA to the sub tour to find better solutions to feed into the main tour. This operation cultivates fitter GENES in a way that mimics biology, the details of which are described in the following section.

2. New algorithm.

The new algorithm is shown in Fig.2.. The parts A and B in the figure are both separately a SGA before part B is intercalated into part A. Part B is the LGA in the algorithm. The two parts form the new algorithm--GGA. Part B performs as a local search operator in the GGA. The GGA is applied to the main tour for searching for global optimal solutions. To obtain the global optimal solutions in the process of the TSP, the GGA may just need to improve the order of a set of cities in the best tour, especially in the latter stages of the process. A set of continuous cities are chosen to create the sub tour in the experiment. The LGA is applied to the sub tour, for finding the local optimal solution to replace the original part chosen from the main tour.

In the LGA, the sub tour is an open tour. The length of the tour is calculated from the start city to the end city, not including the distance between the end city and the start city. Another important point is that all individuals in the population of the LGA have the same start and end cities during the processing. This imperative is for avoiding the main tour becoming longer at the connection points after the recombination of the sub tour.

Local Operations: The tour which consists of n cities is expressed as c=c0, ..., ci, ci+1, ..., cn-1. The distance dci, ci+1 is given for the pair of cities ci and ci+1. All cities are coded using path representation.

1. Randomly choose one main tour $c = c_0, \ldots, c_i, c_j, \ldots, c_{n-1}$ from population of the GGA.
2. Randomly choose one sub tour which contains Ns continuous cities in the main tour c. The sub tour is $c_s = c_i, \ldots, c_j$. The start city is c_i and the end city is c_j. $j-i > 4$. This is the first individual of the LGA. The length of the first individual is set as d_0.
3. Reproduce the first individual to the population size of the LGA. The start city c_i and the end city c_j are kept unchanged. The population of the LGA is created with P_s same individuals.
4. Run the LGA in every generation of the GGA. The start and end cities of all individuals of the LGA are kept unchanged during the processing.
5. The best individual in the population of the LGA is obtained when the LGA stopped. Its length is set as d1.
6. The best individual is recombined back into the main tour to replace the original part.

Let the ratio $Ratio = d_0/d_1$, where the range is $Ratio > 1.0$. The Ratio is discussed again in the next section.

3. Results and Discussions

The main tour which contains n cities is given as $c = c_0, \ldots, c_i, c_{i+1}, \ldots, c_{n-1}$ where the distance between two cities c_i, c_{i+1} is $d(c_i, c_{i+1})$. A sub tour which contains m cities from the main tour is $c = c_0, \ldots, c_k, c_{k+1}, \ldots, c_{m-1}$ where the distance between two cities c_k, c_{k+1} is $d(c_k, c_{k+1})$. Total distances of the main tour and the sub tour are d_m and d_s respectively:

$$d_m = \sum_{i=0}^{n-1} d(c_i, c_{i+1}) \qquad (1)$$

where $c_n = c_0$.

$$d_s = \sum_{k=o}^{m-2} d(c_k, c_{k+1}) \qquad (2)$$

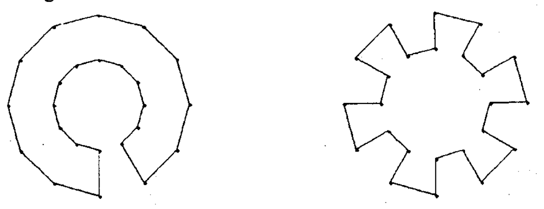
The dm and ds are defined as the Fitness of the individuals. The shorter the distance is, the higher the Fitness is, in other words, the fitness is inversely proportional to the distance. The parameters are shown in Table 1..

Our source code is written in Java and run on a PC (CPU: Pentium III 1.0GHz, RAM: 256MB) with Microsoft Windows2000 Operating System.

Table 1. Parameters in the GGA and the LGA.

	GGA (Main tour)	LGA (Sub-tour)
Population size	100	80
Cities	24	4 24
Generations	1,000 0	10, 20, 30, 40, 50, 60, 70, 80, 90, 100
Crossover rate	75.0%	75.0%
Mutation rate	2.0%	2.5%
Selection	2 shortest individuals replace the 4 longest ones	2 shortest individuals re-place the 4 longest ones

The TSP instance of the double concentric circle, which contains only 24 cities is used in the experiments (Fig.3.). The ratio of the inner radius (R_i) and the outer radius (R_o) is Ri/Ro. If $R_i/R_o < 0.58879$, the optimal tour is C-type (Fig.3.-a). If $Ri/Ro > 0.58879$, the optimal tour is O-type (Fig.3-b).



a. C Type, Ri/Ro<0.58879 b. O Type, Ri/Ro>0.58879

Fig.3. TSP instances.

4.1. Distributions of Optimal Solutions

Fig.4. shows the number of the optimal solutions which is obtained in 20 runs. A peak appears around the mid point, where the number of cities in the sub tour is about half that of the main tour. The Fitness is shown in Fig.5.. The process is stopped at the 1,000th generation. The process converges faster with the increase of the number of cities in the sub tour at the beginning. This happens because there are many more improvable spaces in a big sub tour of the LGA. Stable convergences appears when the city numbers of the sub tour range around half that of the main tour. The result is not satisfied when the LGA performs as the reversion operator in the experiments.

4.2. Computation Time

The GAs usually take a long time to run to reach a good result. Consequently it increases the computation time greatly by combining the LGA into the GGA. The computation time was examined by increasing the city number and the generations in the LGA respectively.

The results are shown in Fig6. and Fig.7.. The computation time linearly became longer with the increase of the number of cities and the generations in

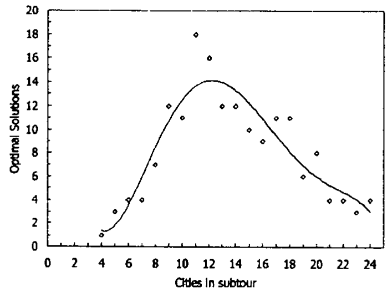


Fig.4. Distribution of optimal solutions

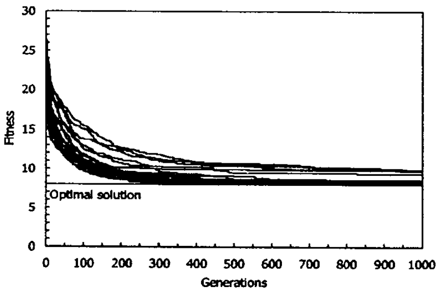


Fig.5. Fitness

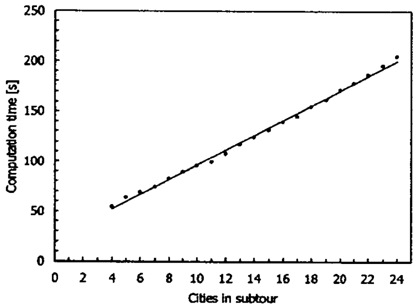


Fig.6. Computation time with cities increase

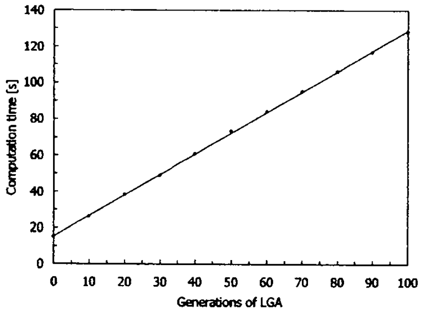


Fig.7. Computation time with generations increase.

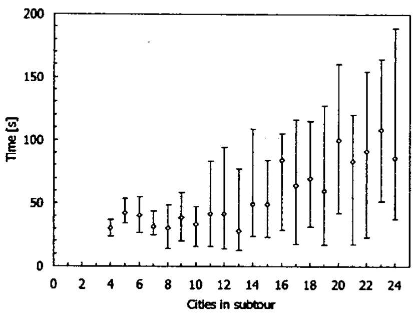


Fig.8. Time of optimal solutions

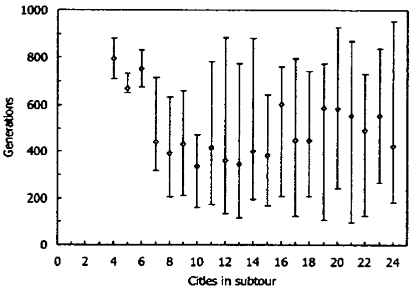


Fig.9. .Generation of optimal solutions

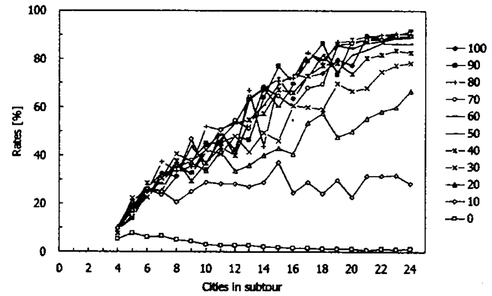


Fig.10. Rates, generations and population sizes.

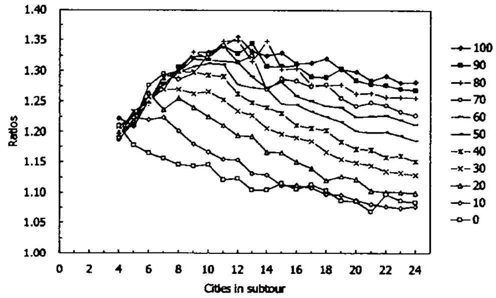


Fig.11. Ratios, generations and population sizes.

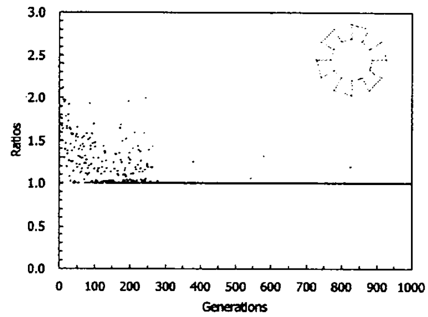


Fig.12. Generation of optimal solution: 252nd

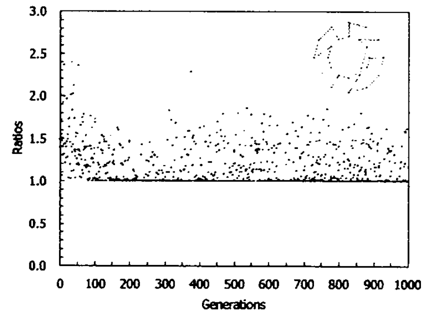


Fig.13. Generation of local solution : 151st

the LGA. Decreasing the number of cities and generations in the LGA took a shorter time, but it might not be sufficient for finding a better sub tour to improve the main tour.

4.3. Time and Generations for obtaining the Optimal Solutions

Fig.8. is the figure of the computational time for obtaining the optimal solutions. There are 3 points on every vertical line; the low, upper, and central points show the shortest, longest, and mean time, respectively for obtaining the optimal solutions in 20 runs. The time taken to find the optimal solutions increases with the number of cities in the sub tour. In Fig.9., the three points on the lines show the earliest, latest, and mean generations for reaching the optimal solutions respectively in 20 runs. The earliest generations appeared when the number of cities in the sub tour ranged around half the number of cities of the main tour. Some optimal solutions appeared early when the sub tours contained more cities. But, it takes a far longer computational time when the number of cities of the sub tour is increased as stated above.

4.4 Recombination Rates and Ratios

Fig.10 is the figure of the recombination rates. We suppose the number of the individuals which were recombined with the LGA is N_r and the number of the individuals which were searched by the LGA is N_s. The recombination rate Rate = Nr/N X 100. The recombination rates increase with the increase of the generations and the number of cities in the sub tours. More generations are advantageous for finding a better sub tour to carry out the recombination, but it takes a longer time to run the LGA. The same trend appears with the changes of the number of cities because there are more improvable spaces in the bigger sub tour. The part chosen from the main tour is easier to improve. Fig.11 is the ratio figure. The ratio Ratio = d_0/d_1 and Ratio > 1.0, where d_0 is the original sub tour from the main tour and d_1 is a sub tour searched to be shorter in the LGA. The peaks of the ratios appear when the number of cities in the sub tour ranged from 8 to 13. The higher ratio means a sub tour was found to be shorter in the LGA than the original part from the main tour. The result is correlative with the distribution of the optimal solutions in Fig.4.

4.5 Dynamics of the LGA

Fig.12 and Fig.13 show the dynamics of the LGA. Fig.12. presents the ratio distributions of a single GGA in which the optimal solution is obtained at the 252nd generation. Because the original part from the GGA is preserved in the LGA and the ratio Ratio = d_0/d_1, the value of the Ratio > 1.0. It indicates that the LGA creates a shorter sub tour and recombines it to the GGA when Ratio > 1.0. The LGA works efficiently and more ratios are bigger than 1.0 before the optimal solution is obtained at the 252nd generation. When the population converges to the optimal solution, the LGA can not create a better sub tour than the original part from the main tour and the ratios become 1.0. Fig.13. shows the ratio distributions of another GGA in which the premature convergence occurs at the 151st generation. There are many points distributed over 1.0 after the premature convergence occurs, which indicates that the LGA still works efficiently even though the GGA reaches the premature convergence.

5. Summary

A local search algorithm based on genetic recombination is discussed in this paper. The LGA acts as a local search operator in the GGA. A good result is presented when the number of cities of the sub tour is set to around half the number of cities of the main tour. Even so, we think it would be reasonable running a small LGA for a big TSP instance because half the number of cities of a big TSP still forms a new big TSP instance. It may be more effective in distributed and parallel processing running the LGA as an island to improve the main tour. This will be discussed in our future works.

References:

[1]. A. Homaifar, S. Guan, and G. E. Liepins, A New Approach to the Traveling Salesman Problem by Genetic Algorithms, in Proceedings of the 5th International Conference on Genetic Algorithms, pp.460--466, Morgan Kaufmann, (1993).

[2]. C. R. Reeves, Genetic algorithms and neighborhood search, in Evolutionary Computing, AISB Workshop, pp.115--130, Leeds, U.K. (1994).

[3]. David E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learnin, Appendix D, Addison-Wesley, Boston, MA, (1989).

[4]. G. Reinelt, The Traveling Salesman: Computational Solutions for TSP Applications, Vol.840 of Lecture Notes in Computer Science, Springer Verlag, Berlin, Germany, (1994).

[5]. Holland, J. H. Adaptation in Natural and Artificial Systems, Univ. of Michigan Press, (1975).

[6]. Keiji Maekawa, Hisashi Tamaki, Hajime Kita, and Yoshikazu Nishikawa, A method for the traveling salesman problem based on the genetic algorithm, Transactions of the Society of Instrument and Control Engineers, Vol.31, No.5, pp.598--605, May (1995) (in Japanese).

[7]. Michael D. Vose, The Simple Genetic Algorithm: foundations and theory, MIT Press, (1999).

[8]. M. Yamamura, T. Ono, and S. Kobayashi, Character-Preserving Genetic Algorithms for Traveling Salesman Problem Journal of The Japanese Society for Artificial Intelligence Vol.7, No.6, pp.1049--1059, (1992) (in Japanese).

[9]. R. M. Brady, Optimization Strategies Gleaned from Biological Evolution, Nature, Vol.317, pp.804--806, textitOperations Research, 28(5), pp.1066--1099, (1985).

[10]. T. G. Bui and B. R. Moon, A New Genetic Approach for the Traveling Salesman Problem, in Proceedings of the First IEEE Conference on Evolutionary Computation, pp.7--12, (1994).