

A Novel Genetic Algorithm and its Application in TSP

Yanbing Liu, Jun Huang

Abstract—A novel genetic algorithm was proposed for the defections of slow convergence and liable to “premature” of traditional GA. It creates crossover and mutation by merging two kinds of heuristics respectively so as to improve the local search ability of GA, and then imports the exterior best-individual set to increase the diversity of the group. The further theoretical analysis and discussion was carried out. simulation results indicated that it can get high-quality solution and consume less running time.

I. INTRODUCTION

Genetic Algorithms (GAs) are stochastic optimization heuristics in which searches in solution space are carried out by imitating the population genetic stated in Darwin’s theory of evolution [1]. It bears advantages of being simple, robust, and easy for parallel processing, but the traditional genetic algorithms in solving large-scale optimization problems perform in low efficiency. It is caused by two reasons; one is the traditional genetic algorithm which only provides a global optimization framework but its local search is poor, and the other is that it is liable to “premature” and very sensitive to selection of initial population. Therefore, most of genetic algorithms are perfected by some other heuristics in the application.

TSP (Traveling Salesman Problem) is a well-known NP-hard optimization problem, requiring the determination of the shortest round trip that passes through a set of cities, each exactly once. TSP raise important issues because various problems in science, engineering, and bioinformatics fields such as vehicle routing, scheduling problems, integrated circuits designs, physical mapping problems, and constructing phylogenetic trees can be formulated as TSP, so the TSP are usually as the typical instance for testing the performance of algorithms. Nowadays, the research of TSP is quite active in that a large number of methods have been developed for solving TSP. For example, [2] refers an approach called Swarm Intelligence, which stands for a class of particle swarm optimization algorithm, regards the city as an intelligent individual, and solves the problem through series of information exchange and learning, and in another case, [3] proposed an improved DNA algorithm for TSP as well as some

other methods for searching the optimal solution, like Simulated Annealing (SA), Neural Network (NN). However, these methods perform quite poorly with this problem. They usually require high execution times, fall into the local-optimum, and are not easy to converge. So, one important research branch is using hybrid genetic algorithm to add it into local search heuristic and diversity maintenance strategy [6-9].

In this paper, we proposed a novel genetic algorithm (New Hybrid Genetic Algorithm, NHGA) responding to the defections of traditional GA, and in terms of characteristics of TSP, introduce the greedy algorithm and Lin-Kernighan heuristic to the crossover and mutation operator respectively, then import the exterior best-individual set to increase the diversity of the group. Theoretical analysis and experimental results show that our proposed methods can get high-quality solution and cost less running time.

II. TSP MODELING

TSP is a typical combinational problem which can be defined as follows:

$$\min \sum_{i \neq j} d_{ij} x_{ij} \quad (1)$$

$$s.t. \sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (2)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n \quad (3)$$

$$\sum_{i,j \in S} x_{ij} \leq |S| - 1, \quad 2 \leq |S| \leq n - 2, \quad (4)$$

$$S \subset \{1, 2, \dots, n\}, x_{ij} \in \{0, 1\}, i, j = 1, \dots, n, i \neq j$$

d_{ij} is the distance between the two cities i and j , $|S|$ stands for the number of elements in set S , equation (2) and (3) is guaranteed the Traveling Salesman visit each city once and only once, formulation (4) constrained the proper subset of Traveling Salesman passed through each city do not exist loops.

III. NHGA FOR TSP

To the TSP-specific issues, NHGA Designation mainly includes four genetic Operators:

A) Selection Operator. We choose the Proportional Model for selection because of simple and convenient. It selected individuals in terms of equation (5).

This work is supported by the National Science Research Foundation of China (No.60702055), Natural Science Foundation of CQUPT(No. A2004-64).

$$P\{T_s(\bar{X}) = X_i\} = \frac{f(X_i)}{\sum_{k=1}^n f(X_k)} \quad (5)$$

B) Crossover Operator. In order to speed up convergence of GA, we imported greedy algorithm formed greedy crossover operator. The main step of greedy crossover described as follows.

Step1: Select N individuals independently from the group $\bar{X}(n)$, it gets the population of $\bar{X}(n) = (X_1, X_2, \dots, X_N)$.

Step2: Select two individuals randomly:

$$X_i(n) = (v_1, v_2, \dots, v_n) \quad (6)$$

$$X_j(n) = (v'_1, v'_2, \dots, v'_n) \quad (7)$$

Step3: Generate a number v_p at random from 1 to n, find the location in (6) and (7) where gene value equal to v_p , then record the position as P_i and P_j respectively.

Step4: Set v_p as the first gene of $X_i(n+1)$, compare the distance of next position $((P_i+1) \bmod n)$ and $((P_j+1) \bmod n)$ gene to the v_p , if $d_i \geq d_j$, set the position $((P_i+1) \bmod n)$ gene which in (7) as the $X_i(n+1)$ next gene and delete the gene in the (6) and (7), or get the next gene from (6).

Step5: Until $X_i(n+1)$ gets all of n genes.

A) Mutation Operator. In NHGA, we merged the Lin-Kernighan heuristic into the mutation operator to compensate the GA's poor local search.

Step1: Select N individuals independently from the group $\bar{X}(n)$, it gets the population of $\bar{X}(n) = (X_1, X_2, \dots, X_N)$.

Step2: Select a individual randomly then execute the Lin-Kernighan heuristic search algorithm.

A) Best-Individual Set operational Operator. Set an exterior best-individual set to store a number of relatively best individuals in current population, then separate the individuals which are kindred. [10] tested two typical TSP instance discovered that, the gap between local optimal solution and global optimal solution is about 20%. That is the reason why we add this operator into NHGA. Here are some details:

Step1: Generated the initial population at random, sort the individual in terms of fitness by descending and keep the first M individuals. ($M \approx 1/5 N$)

Step2: Merged N individuals after mutation and M individuals which reserved in Step1 into a new population which contains $(M+N)$ individuals. We can get Hamming distance between each two individuals in the new population in terms of (8).

$$\|X_i - X_j\| = \sqrt{\sum_{k=1}^N \delta_{ij}^2} \quad (8)$$

$i = 1, 2, \dots, M+N-1$, $j = i+1, \dots, M+N$, δ_{ij} is the indicator function of x_{ik} and x_{jk} , When $x_{ik} = x_{jk}$, $\delta_{ij} = 1$; otherwise,

$$\delta_{ij} = 0.$$

Step3: When $\|X_i - X_j\| > L$, compared the fitness of X_i and X_j , then punish the lower individual.

$$F_{\min(x_i, x_j)} = \text{Penalty} \quad (9)$$

Step4: Sort the individual fitness of new population by descending and reserved former N individuals.

Step5: Turn to Step2, until algorithm satisfied the halt condition.

IV. CONVERGENCE ANALYSIS OF NHGA

Use either SI (MKS) or CGS as primary units. (SI units are strongly encouraged.) English units may be used as secondary units (in parentheses). dimensionally. If you must use mixed units, clearly state the units for each quantity in an equation.

Here are two definitions before we prove the convergence of NHGA.

Definition1^[11]: $F(\bar{X}) = \max\{f(X_i); i \leq N\}$ is the satisfactorily population value of $\bar{X} = (X_1, X_2, \dots, X_N)$.

Definition2^[11]: $M^* = \{\bar{X}; F(\bar{X}) = \max\{f(X); X \in S\}\}$ is the satisfactorily population set.

T_s , T_c , T_m , T_n stands for selection operator, crossover operator, mutation operator and best-individuals set operator respectively, then NHGA can be described as follows:

$$\bar{X}(n) = T(\bar{X}(n-1)) = T_n \circ T_m \circ T_c \circ T_s(\bar{X}(n-1)) \quad (10)$$

Where:

$$P(T_s(\bar{X}(n))) = \begin{cases} \frac{f(Y_1^{(k)}) \cdot f(Y_2^{(k)})}{(\sum_{X \in \bar{X}(n)} f(X))^2} & (Y_1^{(k)}, Y_2^{(k)}) \in \bar{X}(n) \\ 0 & \text{or} \end{cases} \quad (11)$$

$$P(T_c(\bar{X}(n))) = \begin{cases} 2lp_c & Y_1^{(k)} \neq Y_2^{(k)} \\ 1-p_c & Y_1^{(k)} = Y_2^{(k)} \end{cases} \quad (12)$$

$$P(T_m(\bar{X}(n))) = \begin{cases} p_m C_l^6 / l \\ 1-p_m \end{cases} \quad (13)$$

$$P(T_n(\bar{X}(n))) = \begin{cases} \frac{M}{M+1} & \|X_i - X_j\| > L \\ 0 & \text{or} \end{cases} \quad (14)$$

Based on these two definitions we have:

Theorem1 Assume that $\{\bar{X}(n); n \geq 0\}$ is the initial population generated by NHGA, then $\{\bar{X}(n); n \geq 0\}$ is a homogeneous irreducible non-periodic Markov chain with finite state.

Proof There are $n!$ individuals in $S = \{0, 1, \dots, n\}$, and $(n!)^N$ in the population space, S^N is finite. In terms of (8) and the description of the genetic operations, we have T_s , T_c , T_m , is independent of n, so $\bar{X}(n+1)$ is merely related to $\bar{X}(n)$. $\{\bar{X}(n); n \geq 0\}$ is a finite state Markov chain. According to (11), (12), (13), (14) we have:

$$\begin{aligned}
& P\{(T(\bar{X}(n)))_k = X_k(n+1)\} \\
& = \sum_{X_k'(n+1) \in S} \sum_{X_k''(n+1) \in S} \sum_{(Y_1^{(k)}, Y_2^{(k)}) \in S^2} P\{T_s(\bar{X}(n)) = (Y_1^{(k)}(n), \\
& Y_2^{(k)}(n))\} \cdot P\{T_c(Y_1^{(k)}(n), Y_2^{(k)}(n)) = X_k'(n+1)\} \\
& \cdot P\{T_m(X_k'(n+1)) = X_k''(n+1)\} \cdot P\{T_n(X_k''(n+1)) = X_k(n+1)\} \quad (15)
\end{aligned}$$

To any $\bar{X}(n) \in S$, $(Y_1^{(k)}, Y_2^{(k)}) \in S^N$, $X_k'(n+1)$ and $X_k''(n+1)$ always can be draw, we can induce that:

$$P\{T_s(\bar{X}(n)) = (Y_1^{(k)}(n), Y_2^{(k)}(n))\} > 0 \quad (16)$$

$$P\{T_c(Y_1^{(k)}(n), Y_2^{(k)}(n)) = X_k'(n+1)\} > 0 \quad (17)$$

$$P\{T_m(X_k'(n+1)) = X_k''(n+1)\} > 0 \quad (18)$$

To random $X_k'(n+1)$ and $X_k''(n+1)$, we have:

$$P\{T_n(X_k''(n+1)) = X_k(n+1)\} > 0 \quad (19)$$

So

$$P\{(T(\bar{X}(n)))_k = X_k(n+1)\} > 0 \quad (20)$$

According to (11), (12), (13), (14), we know equations (16-20) is independent of n, and we have:

$$P\{T(\bar{X}(n)) = \bar{X}(n+1)\} = \prod_{k=1}^N P\{(T(\bar{X}(n)))_k = X_k(n+1)\} > 0 \quad (21)$$

And it independent of n, so $\{\bar{X}(n); n \geq 0\}$ is a homogeneous irreducible non-periodic Markov chain with finite state.

Theorem2: $\{\bar{X}(n); n \geq 0\}$ converges to

$M_0^* = \{\bar{Y} = (Y_1, Y_2, \dots, Y_N)\}$ with probability one.

Proof assume $P_n(\bar{X}, \bar{Y}) = P\{\bar{X}(n+1) = \bar{Y} / \bar{X}(n) = \bar{X}\}$

$$P(n) = (P_n(\bar{X}, \bar{Y}); \bar{X}, \bar{Y} \in S^N)$$

$$P(\infty) = \lim_{n \rightarrow \infty} P_n(\bar{X}, \bar{Y}) = P\{T_n \circ T_m \circ T_c \circ T_s(\bar{X}) = \bar{Y}\}.$$

According to the proof of Theorem1, it certainly existed a stationary distribution π makes that:

$$\lim_{n \rightarrow \infty} P\{\bar{X}(n) = \bar{Y} / \bar{X}(0) = X_0\} = \begin{cases} \pi(\bar{Y}), & Y \in M_0^* \\ 0 & Y \notin M_0^* \end{cases} \quad (22)$$

Obviously, $P(\infty)$ have an unique irreducible non-periodic positive recurrence M_0^* , and S/M_0^* is a non-recurrence, so

$\{\bar{X}(n); n \geq 0\}$ is strongly ergodic, to an arbitrary initial state $\bar{X}(0) = X_0$, we have

$$\lim_{n \rightarrow \infty} P\{\bar{X}(n) = \bar{Y} / \bar{X}(0) = X_0\} = \pi(\bar{Y}) \quad \text{and} \quad \sum_{\bar{Y} \in M} \pi_\infty(\bar{Y}) = 1, \quad (23)$$

therefore, we get:

$$\lim_{n \rightarrow \infty} P\{\bar{X}(n) = \bar{Y} / \bar{X}(0) = X_0\} = \sum_{\bar{Y} \in M} \pi_\infty(\bar{Y}) = 1 \quad (23)$$

From the proof of above theorems, we can get the population generated by NHGA are convergent.

V. SIMULATIONS AND PERFORMANCE ANALYSIS

Table1 Parameters of GA

Parameter	P_c	P_m	POP	GEN	BEST_SET
Value	0.65	0.1	500	1000	100

We run NHGA on PC Pentium D 2.8G and 512M DDR, Table1 shows the parameters of GA.

The difference between HGA and NHGA is selection operator which is elite model in HGA. Similarly, we could prove that HGA can converge to the satisfactorily population set with probability one.

Table2 shows the result of 10 typical test instances in TSPLIB [13], where Efficiency stands for the ratio of average result which tested 20 times and global optimum. We can see from the Table2 that, as to the former 5 test instances, HGA can converge to the satisfactorily solution very quickly. It is because that time complexity is $O(mn^2)$, when a is small, m is the main factor was considered.

Experiment2: Validity test on increasing population diversity of NHGA.

Table 2. Experimental Result of HGA

TSP	Opt.	Best Result	Average Result	Efficiency
Burma14	33.23	33.23	33.23	1.00
Bayg29	9074	9074	9074	1.00
Att48	33524	33524	33524	1.00
Eli51	426	426	426	1.00
St70	675	675	675	1.00
Eli101	629	686	700	1.11
Ch130	6110	6537	6645	1.09
Ch150	6528	7035	7228	1.11
Tsp225	3919	4496	4573	1.16
Pcb442	50778	57731	58961	1.16

Figure1 shows the NHGA performance of four Eli101, Ch130, TSP225, Pcb442 in the latter 5 test instances compared with HGA. From the curve we can see that NHGA can improve the population fitness more or less, it means that NHGA can increase population diversity and escape the local optimum effectively especially in Eli101. The solution which gets from NHGA is very close to the global optimum. It also can be seen that population diversity is not only decided by the genetic operations but also the scale of initial population.

Figure 2 validates our method correctness again, Limited to experimental condition, we set 1000 initial individuals of TSP225 then run this instance. As the fitness shown, population fitness will increase when set a large scale initial population, and we could predict that (because it will overflow when we set 2000 initial population) NHGA can always get a solution which is very close to the optimum.

VI. CONCLUSIONS

This paper starts from the two drawbacks of traditional GA: convergent slowly and easy to “premature”. A new class of genetic algorithm is given through improvement. By testing typical TSP instance, NHGA can converge very quickly and escape local optimum effectively. Theoretical analysis of experimental data and simulation results are verified the feasibility and effectiveness of the algorithm, and It provides a new way of designation of genetic algorithms in its application.

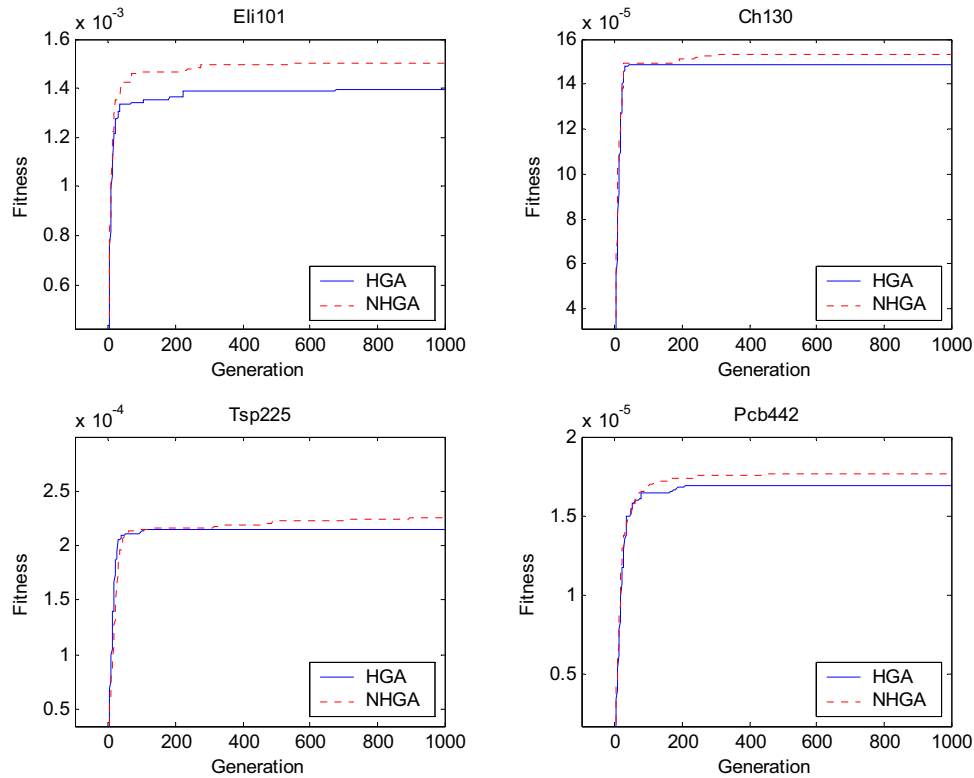


Figure 1. NHGA Compared with HG

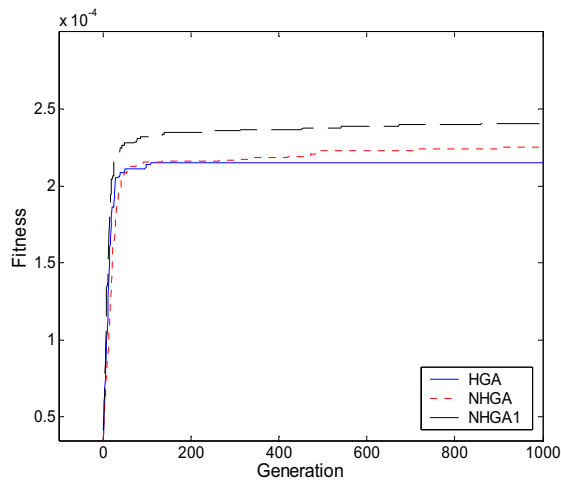


Figure 2. TSP225 with 1000 initial population

REFERENCES

- [1] J.H.Holland, *Adaptation in Natural and Artificial System*, University of Michigan Press, USA, 1975.
- [2] X.He, Y.Zhu, H.Hu. "A new algorithm for TSP based on swarm intelligence", *Proceedings of the 6th World Congress on Intelligent Control and Automation*, 2006, pp. 3241 - 3244

- [3] K.Mehdizadeh, M.A.Nekou, and K.Sabahi. "A modified DNA-computing algorithm to solve TSP", *2006 IEEE International Conference on Mechatronics*, 2006, pp. 65 - 68.
- [4] C.S.Jeong and M.H.Kim, "Fast parallel simulated annealing for traveling salesman problem", *International Joint Conference on Neural Networks*, 1990, pp. 947 - 953.
- [5] E.Gelenbe, V.Koubi and F.Pekergin, "Dynamical random neural network approach to the traveling salesman problem", *Systems Engineering in the Service of Humans Conference Proceedings*, 1993, pp.630- 635.
- [6] H.D.Nguyen, Kunihiro, Yamamori and Moritoshi Yasunaga, "Implementation of an effective hybrid GA for large-scale traveling salesman problems", *IEEE Transactions On Systems, Man, And Cybernetics*, 2007, 37(1), pp. 92-99.
- [7] P.Gang, I.Ilimura, and S.Nakayama, "A multiple heuristic search algorithm for solving traveling salesman problem", *Proceedings of the Fourth International Conference on Parallel and Distributed Computing, Applications and Technologies*, 2003, pp. 779 - 783.
- [8] C.F.Tsai, C.W.Tsai and C.C.Tseng, "A new approach for solving large traveling salesman problem", *Proceedings of the 2002 Congress on Evolutionary Computation*, 2002, pp. 1636 - 1641.
- [9] J.Firbes and Burkowski, "Proximity and priority: applying a gene expression algorithm to the traveling salesperson problem", *International Parallel and Distributed Processing Symposium* 2003.
- [10] P.Zhou and J.Gu, "A Multi-level Reduction Algorithm in TSP", *Journal of Software*, 2003, 14(1), pp. 35-42.
- [11] W.X.Zhang and Y.Liang. *Mathematical Foundation of Genetic Algorithm*, Xi'an Jiaotong University, Xi'an, 2002.
- [12] S.Lin and B.W.Kernighan, "An effective heuristic algorithm for the traveling-salesman problem", *Operations Research*, 1973, 21(2), pp. 498 - 516.
- [13] <http://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB9>