

A Novel Hybrid Intelligence Algorithm for Solving Combinatorial Optimization Problems

Wu Deng^{1,2,3,4*}, Han Chen^{1,2}, and He Li^{1,4}

¹Software Institute, Dalian Jiaotong University, Dalian, China

²The Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou, China

³The Artificial Intelligence Key Laboratory of Sichuan Province, Sichuan University of Science and Engineering, Zigong, China

⁴Guangxi Key Laboratory of Hybrid Computation and IC Design Analysis, Guangxi University for Nationalities, Nanning, China
dw7689@gmail.com, 949261902@qq.com, Heli2015win@hotmail.com

Abstract

The ant colony optimization (ACO) algorithm is a new heuristic algorithm that offers good robustness and searching ability. With in-depth exploration, the ACO algorithm exhibits slow convergence speed, and yields local optimization solutions. Based on analysis of the ACO algorithm and the genetic algorithm, we propose a novel hybrid genetic ant colony optimization (NHGAO) algorithm that integrates multi-population strategy, collaborative strategy, genetic strategy, and ant colony strategy, to avoid the premature phenomenon, dynamically balance the global search ability and local search ability, and accelerate the convergence speed. We select the traveling salesman problem to demonstrate the validity and feasibility of the NHGAO algorithm for solving complex optimization problems. The simulation experiment results show that the proposed NHGAO algorithm can obtain the global optimal solution, achieve self-adaptive control parameters, and avoid the phenomena of stagnation and prematurity.

Category: Smart and intelligent computing

Keywords: Genetic algorithm; Ant colony optimization algorithm; Multi strategies; Hybrid evolutionary algorithm; Combinatorial optimization problem

I. INTRODUCTION

Many problems in industry, agriculture, national defense, information, and other areas can be transformed into combinatorial optimization problems [1]. When intelligent optimization algorithms are used to solve such combinatorial optimization problems, these algorithms reveal their advantages and disadvantages. In response to specific complex problems, many researchers have proposed hybrid intelligent algorithms based on fusing different intelligent optimization algorithms, in order to realize the

complementarity of these advantages and value-added information [2, 3]. Thereby, these complex optimization problems can obtain better solutions. At present, the intelligent optimization algorithms mainly include the evolution algorithm [4] and the swarm intelligence optimization algorithm [5]. The genetic algorithm (GA) is a highly parallel, random, and adaptive search algorithm, which employs the mechanism of natural selection and natural genetics in living nature. It includes selection operators, crossover operators, and mutation operators. It can search for the optimal target by using the population,

Open Access <http://dx.doi.org/10.5626/JCSE.2014.8.4.199>

<http://jcse.kiise.org>

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 6 July 2014; Revised 3 November 2014; Accepted 16 November 2014

*Corresponding Author

and emphasizing global search ability. But it is inadequate in terms of local meticulous search ability. Meanwhile, the ant colony optimization (ACO) algorithm strengthens local search ability by the positive feedback of the pheromone. But it is not strong in terms of global search ability. The imbalance between the global search and local search could cause the phenomena of prematurity and stagnation. The GA and ACO algorithms are intelligent optimization algorithms that simulate biological evolutionary processes. In essence, they take on the characteristics of parallelism, robustness, and self-organization; so that they do not rely on a mathematical model to solve complex optimization problems, and instead rely on the ability of the population to search for the optimal solution. So if the GA and ACO algorithms can be combined to overcome their own shortcomings, while avoiding the prematurity and stagnation phenomena, this will enable the maximum extent of collaborative optimization.

In this paper, we analyze the dynamic characteristics, mechanisms, optimization strategies, and convergence of the GA and ACO algorithms. We introduce adaptive control strategy and multi-population strategy into the GA and ACO algorithms, in order to propose a novel hybrid genetic ant colony optimization (NHGAO) algorithm, based on combining these strategies. The NHGAO algorithm takes advantage of the merits of the GA and ACO algorithms to solve complex optimization problems, in order to fully reflect the whole optimization ability of the NHGAO algorithm, dynamically balance the contradiction between the convergence speed and search range, overcome the low efficiency of the GA at a certain stage, and the initial pheromone distribution of the ACO algorithm.

II. RELATED WORKS

In recent years, in response to the existing inadequacy of the GA and ACO algorithms, there have been many attempts to use new approaches or strategies to solve combinatorial optimization problems, in order to obtain better solutions. Lee et al. [6] proposed a novel algorithm consisting of the GA with ACO for multiple sequence alignment. In the proposed GA-ACO algorithm, the GA is performed to provide a diversity of alignments. ACO is performed to move out of local optima. Nemati et al. [7] proposed a novel feature selection algorithm that combines GAs and ACO for faster and better search capability. The hybrid algorithm makes use of the advantages of both ACO and GA methods. Sheikhan and Mohammadi [8] proposed two hybrid models for short-term load forecasting. These models use the ACO and a combination of the GA and ACO (GA-ACO) for feature selection, and a multi-layer perceptron for hourly load prediction. Shuang et al. [9] proposed a hybrid PS-ACO algorithm—the ACO algorithm modified by the particle swarm optimization (PSO) algorithm. The pheromone updating rules of ACO

are combined with the local and global search mechanisms of the PSO. On the one hand, the search space is expanded by local exploration; on the other hand, the search process is directed by global experience. Dong et al. [10] proposed a new hybrid algorithm, or the cooperative genetic ant system (CGAS), to deal with the travelling salesman problem (TSP). This new approach combines both the GA and ACO together in a cooperative manner to improve the performance of the ACO for solving the TSP. Saenphon et al. [11] proposed a new evolutionary optimization algorithm based on the actual manifold of objective function and fast opposite gradient search in order to improve the accuracy and speed of solution finding. The first phase searches the best candidate solutions by using our Fast Opposite Gradient Search on the manifold of objective function. The second phase applies ACO to improve the candidate solutions.

III. THE INTELLIGENT OPTIMIZATION ALGORITHMS

A. Genetic Algorithm

The GA [12] is a class of population-based stochastic search techniques that solve problems by imitating processes observed during natural evolution. It is based on the principle of the survival and reproduction of fitness. The GA is a parallel iterative algorithm with certain learning ability, which repeats the evaluation, selection, crossover and mutation operator after initialization, until the stopping criteria are reached. In the GA, a population of candidate solutions evolves. Each solution in the candidate solutions is encoded as a binary string, called a chromosome. A fitness function is used to evaluate the fitness value of each chromosome. A real-coded GA is a

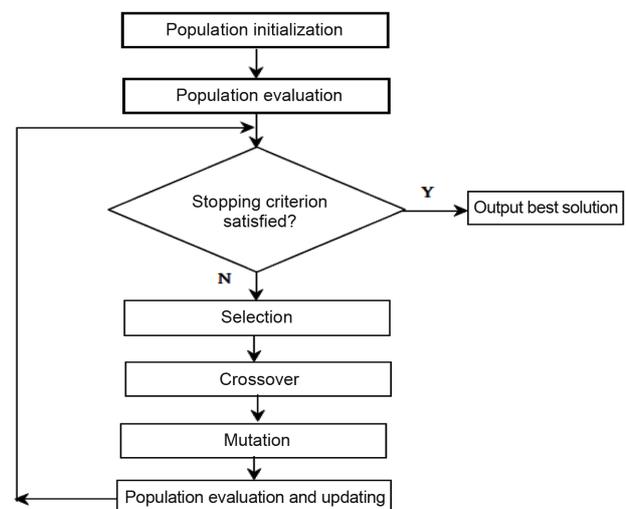


Fig. 1. Searching procedure of the genetic algorithm.

genetic algorithm representation that uses a vector of floating-point numbers, instead of 0 and 1, to implement chromosome encoding. With some modifications of the genetic operators, the real-coded GA has better performance than the binary-coded GA. The crossover operator of a real-coded GA is performed by borrowing the concept of convex combination. The random mutation operator is used to change the gene with one random number in the problem's domain, assuming that we employ the GA to search for the largest fitness value with a given fitness function, as Fig. 1 shows.

B. Ant Colony Optimization Algorithm

The ACO [13] algorithm was introduced by Marco Dorigo. It is a branch of a newly developed form of artificial intelligence called swarm intelligence, which studies the emergent collective intelligence of groups of simple agents. The ACO is a metaheuristic inspired by the behavior of real ants in their search for the shortest path to food. The ACO consists of a number of cycles (iterations) of solution construction. During each iteration, a number of ants construct complete solutions by using heuristic information, and the collected experiences of previous groups of ants. These collected experiences are represented by the pheromone trail, which is deposited on the constituent elements of a solution [14-16]. Small quantities are deposited during the construction phase, while larger amounts are deposited at the end of each iteration in proportion to the solution quality. Pheromone can be deposited on the components and/or the connections used in a solution depending on the problem. Fig. 2 illustrates the ACO procedure.

IV. MULTI-STRATEGY ADAPTIVE CONTROL METHOD

A. Self-Adaptive Control Parameters of the GA

In order to balance the search range and search ability of the GA, many researchers have proposed improved

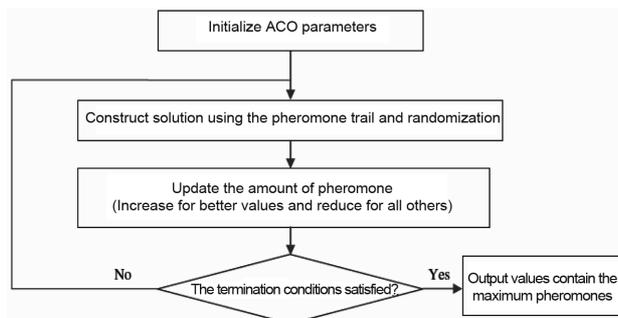


Fig. 2. Flow chart of the ant colony optimization (ACO).

GAs. For example, Srinivas proposed an adaptive GA, which uses the difference between the average fitness value of the population and the fitness value of the optimal individual to determine the diversity of the population. However, this algorithm cannot promptly reflect the premature convergence of an individual. If the difference between the small fitness value of the individual and the average fitness value of the population is large, then the algorithm will mistakenly believe that the population did not discover a premature convergence. Also, the algorithm cannot escape from a local optimal solution to reduce the search optimization performance. If the fitness value of the individual is greater than the average fitness value of the population ($\overline{f(t)}$), the fitness values of these individuals will redo the method in order to obtain the average value ($\overline{f(t)}$). Define $\Phi = f_{GA}^{\max}(t) - \overline{f(t)}$, which represents the premature convergence extent of the population. Then:

$$P_c(t) = \begin{cases} k_1 \frac{f_{GA}^{\max}(t) - f'_{GA}(t)}{f_{GA}^{\max}(t) - \overline{f(t)}}, & f'_{GA}(t) \geq \overline{f(t)} \\ k_3, & f'_{GA}(t) \leq \overline{f(t)} \end{cases} \quad (1)$$

$$P_m(t) = \begin{cases} k_2 \frac{f_{GA}^{\max}(t) - f_{GA}(t)}{f_{GA}^{\max}(t) - \overline{f(t)}}, & f_{GA}(t) \geq \overline{f(t)} \\ k_4, & f_{GA}(t) \leq \overline{f(t)} \end{cases} \quad (2)$$

where the coefficient values of $k_1, k_2, k_3,$ and k_4 are less than or equal to 1. $f'_{GA}(t)$ is the larger fitness function value in two crossover individuals. $f_{GA}(t)$ is the used function value of a mutation individual.

B. Self-Adaptive Control Parameters of the ACO

In the transition of the ant colony, the value of $q(t)$ is a consistently distributed random number on the $[0,1]$ interval. Thus the diversity of the solution increases. To a certain extent, the trend of falling into the local optimum weakens. But $q(t)$ is difficult to control because of the random number. So this paper uses a uniformly distributed random number on the $[0,1]$ interval.

$$q(t+1) = \begin{cases} q(t)/\mu, & q(t) \in (0, \mu) \\ (1-q(t))/(1-\mu), & q(t) \in (\mu, 1) \end{cases} \quad (3)$$

where $\mu \in (0, 1)$. Due to the existing pheromone evaporation system $\rho(t)$, if the scale of the problem is large, the amount of pheromone on the not being searched path will gradually reduce to close to zero because the pheromone evaporation coefficient $(1-\rho)$ is adopted. This states that a path that reduces the global search ability will not be searched. If the value of ρ is too large with an increase of the pheromone of the solution, the previously searched solution is more likely to be selected, which will affect the global search ability of the algorithm. So the value of

ρ should reduce in order to improve the global search ability. But this will reduce the convergence speed. In response to the above shortcomings, we can use the adaptive method to control the value of ρ . The specific realized approach is: The initial value is $\rho = 0.999$. When the optimal value does not obviously change in T cycle times, the value of ρ will reduce to:

$$\rho(t+1) = \begin{cases} (0.9 + rand() / 10 * (RAND_MAX + 1)) \rho(t), & \rho(t+1) \geq \rho_{\min} \\ \rho_{\min}, & \text{otherwise} \end{cases} \quad (4)$$

where ρ_{\min} is the minimum value of ρ in order to prevent reducing the convergence of the algorithm, because of the too small value of ρ . $rand()$ is a random function. The method of the adaptive changing ρ value effectively improves the global search ability of the algorithm under a certain search speed.

V. THE IDEA AND A DESCRIPTION OF THE NHGAO ALGORITHM

The basic idea of the NHGAO algorithm is that in each cycle of the NHGAO algorithm, the GA and ACO algorithms guide each other by updating the global optimization solution. In the GA, a population is regarded as an ant colony, and then one individual in the population is regarded as an ant in the ant colony. The obtained optimal solution by the crossover and mutation operations is used to describe the selected action of the path of the ant. The selection operation based on fitness evaluation function can preserve the relation of each generation among these ants. In the ACO algorithm, each ant selects one path according to the heuristic function and the consistency of the pheromone. The ant colony uses the pheromone that is left to achieve the collaborative purpose. Each ant may be regarded as an individual in the GA. The ant colony of each generation may be regarded as the population of each generation in the GA. The ACO algorithm emphasizes the search process of the individual. The collaborative work of the ant colony is obtained by using positive feedback operation of the pheromone that is left. The NHGAO algorithm obtains the global search ability of the GA and the local search ability and convergence speed of the ACO algorithm by the multiple cycles. So the two algorithms combine and complement each other. The NHGAO algorithm functions as an adaptive collaborative optimization algorithm with better time efficiency and solving efficiency.

The description of the NHGAO algorithm is shown:

Step 1: Generate the initial population, and initialize the other parameters. These parameters include the initial population (M_G), number of ants, crossover probability (P_c), mutation probability (P_m), and iteration (T_{max}).

Step 2: Calculate the fitness value of each individual, by comparing and updating the optimal fitness value. The

average fitness value is calculated according to the following formula:

$$\bar{F}_G(t) = \frac{1}{M_G} \sum_{G^i=1}^{M_G} F_G^i(t) \quad (5)$$

$$\bar{F}_A(t) = \frac{1}{M_A} \sum_{A^i=1}^{M_A} F_A^i(t) \quad (6)$$

Step 3: Adaptively control the parameters of the GA in order to obtain the crossover probability $P_c(t)$ and mutation probability $P_m(t)$ in this iteration.

Step 4: Set the initial iteration $t = 0$.

Step 5: Execute mutation operation of the GA. Each parent individual generates a sub-individual following the generating method:

$$x'_i = (x_i + \lfloor n! |N(0, 1)| \rfloor) \bmod n! \quad (7)$$

Step 6: Obtain several optimal solutions. The obtained fitness values of the algorithm are compared in order to the global optimal value, $gBest$. If the accuracy meets the requirement, the algorithm goes into step 5 to continue running, otherwise looping through steps 2 to 5, until reaching the set maximum number of iterations.

Step 7: Use the optimal solution to initialize the pheromone on each path in order to obtain the value of the pheromone of each path at the same time.

Step 8: Each ant in the ACO algorithm selects the next access node according to expressions (8) and (9).

$$j = \begin{cases} \arg \max \{ [\tau_{ij}(t)] [\eta_{ij}(t)]^\beta \}, & q(t) \leq q(\lambda(t)) = \lambda(t) / N \\ S, & \text{otherwise} \end{cases} \quad (8)$$

$$p_k(i, j) = \begin{cases} \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta} & j \in J_k(i) \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

Step 9: Update the pheromone according to expression (10).

$$\tau(i, j) = (1 - \rho) \times \tau(i, j) + \sum_{k=1}^m \Delta \tau_k(i, j) \quad (10)$$

Step 10: Calculate the completed length of one visited path of each ant, and the fitness value of the solution. Record the optimal solution $S_{best}(t)$ of this iteration. If the value of $S_{best}(t)$ is better than S_{best} , then $S_{best} = S_{best}(t)$. Otherwise return to step 8.

Step 11: Adaptively control the parameters of the ACO algorithm to update the global pheromone.

Step 12: Calculate and record the average value $\bar{F}_A(t)$ of the obtained solutions of the ACO algorithm in this iteration.

Step 13: Calculate the fitness function value and the

average value of the NHGAO algorithm according to the expression. The fitness value of the obtained individual that is greater than the value of $\bar{F}(t)$ is determined to replace the lower fitness value of the individual in the initial population.

Step 14: Calculate the fitness function values of all solutions. If the obtained optimal solution is better than the historical optimal solution, the obtained optimal solution will replace the historical optimal solution. Otherwise, return to step 12.

Step 15: Calculate the average value of the fitness function ($F(t+1)$). If the termination condition is met, output the history optimal solution. Otherwise $t = t + 1$, and return to step 5.

VI. THEORETICAL ANALYSIS OF THE NHGAO ALGORITHM

The parent-individuals on the coding space generate sub-individuals according to expression (7). When the mutation value is larger, the Hamming distance between parent-individuals and sub-individuals is even greater for two full permutations on the corresponding problem space. When the mutation value is small, the Hamming distance between parent-individuals and sub-individuals is smaller with high probability.

The character accords with the solving idea of the evolutionary algorithm. The NHGAO algorithm achieves global optimization with a larger mutation value, and local optimization with a smaller mutation value. Accordingly, this explains the effectiveness of the NHGAO algorithm. The individuals are $n-1$ natural binary. After a single point crossover is executed, the legal $n-1$ natural binary is obtained. So the crossover operation cannot generate invalid individuals.

VII. EXPERIMENTAL RESULTS AND ANALYSIS

In order to test the performance of the NHGAO algorithm, we selected several instances from TSPLIB (<http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/>) to test the algorithm. We used MATLAB 2010 to test these instances. According to the characteristics of TSPLIB, the distance between any two cities is calculated by the Euclidian distance. At the same time, the GA, ACO, and NHGAO algorithms are selected to analyze and compare their optimized performances. The initial parameters of the algorithms are selected after thorough testing. In the simulation experiments, the alternative values were tested and modified for some functions in order to obtain the most reasonable initial values of these parameters. In the simulation experiments, this paper considers the random factors of the GA, ACO, and NHGAO algorithms. So for each TSP, the best value and the average value are com-

Table 1. The results of the simulated experiments

Inst.	GA		ACO		NHGAO	
	Best	Avg.	Best	Avg.	Best	Avg.
eil51	443.25	465.16	431.32	439.13	426	429.03
berlin52	7623.2	7678.4	7587.5	7604.3	7543.3	7545.2
rat99	1312.2	1338.7	1265.8	1297.2	1217.1	1225.5
ch130	6120.3	6194.3	6149.3	6175.3	6116.7	6121.5
u159	42834	42966	42491	42508	42388	42410
kroA200	29868	29984	29511	29525	29383	29436
pr299	48739	48897	48400	48485	48248	48311
rd400	15764	15932	15461	15520	15334	15393
rat575	6989.1	7098.5	6916.3	6997.3	6856.4	6903.4
rat783	9102.3	9169.8	8998.2	9016.6	8948.2	8979.1
pr1002	265669	266149	264141	265680	263282	264187
pcb3038	143341	144902	141095	141955	140593	141043

GA: genetic algorithm, ACO: ant colony optimization, NHGAO: novel hybrid genetic ant colony optimization.

pared. These algorithms are executed on each TSP instance with ten trials, and Table 1 shows the results of the simulated experiments.

As can be seen from Table 1, for the 12 TSP instances with GA, ACO, and NHGAO algorithms, the values of best and average of the NHGAO algorithm are the best for all experiments. In addition, for TSP instances eil51, the proposed NHGAO algorithm can find the best-known solution 426. In particular, for TSP instances berlin52 and kroA200, the new best-known solutions 7543 and 29383 approach the best-known solutions 7542 and 29368. For larger scale instances, Table 1 shows that the average results of the proposed NHGAO algorithm are better than those of other methods.

In order to validate the effectiveness of the proposed NHGAO algorithm, the NHGAO algorithm is compared with other methods [17-19], for 15 TSP benchmark instances from TSPLIB with city scales from 51 to 1655. Table 2 shows a comparison of the experimental results.

From Table 2, we can see that the best solution found of the NHGAO algorithm is better than those of the other three algorithms for the data sets eil51, ch130, ch1150, kroA200, kroB200, lin318, rat575, rat783, and d1655. The average solution found of the NHGAO algorithm is better than those of the other three algorithms for the data sets eil76, rad100, eil101, lin105, ch1150, kroB150, kroA200, kroB200, lin318, rat575, rat783, fl140, and d1655. Fig. 3 shows a comparison of the best solution and the average solution to the best-known solution, for the different methods. Table 2 and Fig. 3 reveal that the NHGAO algorithm shows better optimization performance than the other methods in the mass for each TSP.

Table 2. Comparison of the experimental results

Inst.	Angeniol et al. [17]		Pasti and de Castro [18]		Chen and Chien [19]		NHGAO	
	Best	Avg.	Best	Avg.	Best	Avg.	Best	Avg.
eil51	432	442.90	433	440.57	427	427.27	426	427.35
eil76	554	563.20	552	562.27	538	540.20	539.37	540.18
rad100	8088	6842.8	7947	8253.9	7910	7987.6	7910	7934.2
eil101	655	665.93	640	655.57	630	635.23	630	634.93
lin105	14999	16111	14379	14477	14379	14406	14382	14395
ch130	6265	6416.8	6203	6307.2	6141	6205.6	6114.3	6123.7
ch1150	6634	6842.80	6631	6751	6528	6563.7	6528.0	6539.9
kroB150	27886	29405	26342	26806	26130	26448	26139	26176
kroA200	31669	33228	30144	30416	29383	29739	29382	29437
kroB200	32351	33838	29703	30287	29541	30035	29503	29514
lin318	44869	45839	43154	43923	42487	43003	42267	42363
rat575	8107	8301.8	7090	7173.6	6891	6933.9	6859.4	6901.7
rat783	10532	10722	9316	9387.6	8988	9079.2	8946.4	8978.3
fl140	20649	21175	20558	20743	20593	21350	20686	20779
d1655	68875	71168	67459	68046	64151	65621	63618	64131

NHGAO: novel hybrid genetic ant colony optimization.

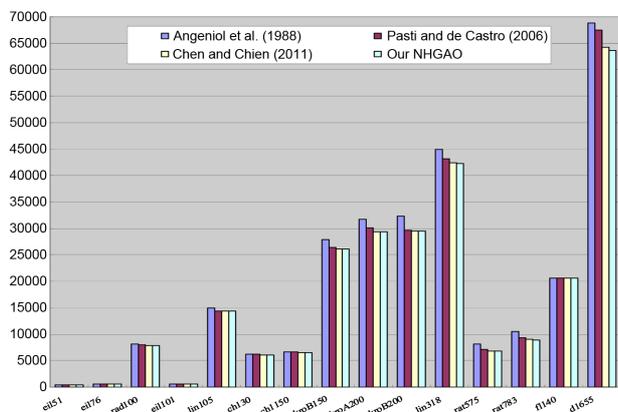


Fig. 3. Comparison of the average tour.

VIII. CONCLUSION

In response to the prematurity and stagnation phenomena of the GA and ACO, we analyze the operating characteristics of the GA and ACO, and introduce an adaptive control strategy and a multi-population strategy into the GA and ACO in order to propose an NHGAO algorithm based on these strategies. The NHGAO algorithm uses the characteristics of random search and global convergence of the GA, and the parallelism, positive feedback mechanism and high solving efficiency of the ACO algorithm to update the global optimal solution. The multi-

population strategy can integrate and share the good genes of each population, improve the population, and accelerate the evolution speed of the population. At the same time, the diversity of the population, convergence speed, and determining function are defined to evaluate the current running state of the algorithm. The running algorithm is automatically adjusted according to the evaluation results of the algorithm in order to achieve genetic and ant colony coevolution. The simulation results fully demonstrate that the proposed NHGAO algorithm accelerates the convergence speed, improves the results of the search problem, and suppresses the prematurity and stagnation phenomena.

ACKNOWLEDGMENTS

The authors would like to thank all the reviewers for their constructive comments. This research was supported by the National Natural Science Foundation of China (U1433124,51475065), the Open Project Program of the Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1326), the Open Project Program of Artificial Intelligence Key Laboratory of Sichuan Province (Sichuan University of Science and Engineering) (2014RYJ02,2014RYJ01), the Open Project Program of the State Key Laboratory of Mechanical Transmissions (Chongqing University) (SKLMT-KFKT-201416), and the Open Project Program of Guangxi

Key laboratory of hybrid computation and IC design analysis (HCIC201402).

REFERENCES

1. C. Blum and A. Roli, "Metaheuristics in combinatorial optimization: overview and conceptual comparison," *ACM Computing Surveys*, vol. 35, no. 3, pp. 268-308, 2003.
2. Y. Zheng and B. Liu, "Fuzzy vehicle routing model with credibility measure and its hybrid intelligent algorithm," *Applied Mathematics and Computation*, vol. 176, no. 2, pp. 673-683, 2006.
3. E. Corchado, A. Abraham, and A. de Carvalho, "Hybrid intelligent algorithms and applications," *Information Sciences*, vol. 180, no. 14, pp. 2633-2634, 2010.
4. H. C. Kuo and C. H. Lin, "Cultural evolution algorithm for global optimizations and its applications," *Journal of Applied Research and Technology*, vol. 11, pp. 510-522, 2013.
5. P. Tarasewich and P. R. McMullen, "Swarm intelligence: power in numbers," *Communications of the ACM*, vol. 45, no. 8, pp. 62-67, 2002.
6. Z. J. Lee, S. F. Su, C. C. Chuang, and K. H. Liu, "Genetic algorithm with ant colony optimization (GA-ACO) for multiple sequence alignment," *Applied Soft Computing*, vol. 8, no. 1, pp. 55-78, 2008.
7. S. Nemati, M. E. Basiri, N. Ghasem-Aghaei, and M. H. Aghdam, "A novel ACO-GA hybrid algorithm for feature selection in protein function prediction," *Expert Systems with Applications*, vol. 36, no. 10, pp. 12086-12094, 2009.
8. M. Sheikhan and N. Mohammadi, "Neural-based electricity load forecasting using hybrid of GA and ACO for feature selection," *Neural Computing and Applications*, vol. 21, no. 8, pp. 1961-1970, 2012.
9. B. Shuang, J. Chen, and Z. Li, "Study on hybrid PS-ACO algorithm," *Applied Intelligence*, vol. 34, no. 1, pp. 64-73, 2011.
10. G. Dong, W. W. Guo, and K. Tickle, "Solving the traveling salesman problem using cooperative genetic ant systems," *Expert Systems with Applications*, vol. 39, no. 5, pp. 5006-5011, 2012.
11. T. Saenphon, S. Phimoltares, and C. Lursinsap, "Combining new fast opposite gradient search with ant colony optimization for solving travelling salesman problem," *Engineering Applications of Artificial Intelligence*, vol. 35, pp. 324-334, 2014.
12. J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, Ann Arbor, MI: University of Michigan Press, 1975.
13. M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73-81, 1997.
14. T. Kotzing, F. Neumann, H. Roglin, and C. Witt, "Theoretical analysis of two ACO approaches for the traveling salesman problem," *Swarm Intelligence*, vol. 6, no. 1, pp. 1-21, 2012.
15. K. Y. Lee and F. F. Yang, "Optimal reactive power planning using evolutionary algorithms: a comparative study for evolutionary programming, evolutionary strategy, genetic algorithm, and linear programming," *IEEE Transactions on Power Systems*, vol. 13, no. 1, pp. 101-108, 1998.
16. D. Mester, Y. Ronin, D. Minkov, E. Nevo, and A. Korol, "Constructing large-scale genetic maps using an evolutionary strategy algorithm," *Genetics*, vol. 165, no. 4, pp. 2269-2282, 2003.
17. B. Angeniol, G. de La Croix Vaubois, and J. Y. Le Texier, "Self-organizing feature maps and the travelling salesman problem," *Neural Networks*, vol. 1, no. 4, pp. 289-293, 1998.
18. R. Pasti and L. N. De Castro, "A neuro-immune network for solving the traveling salesman problem," in *Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN2006)*, Vancouver, Canada, 2006, pp. 3760-3766.
19. S. M. Chen and C. Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques," *Expert Systems with Applications*, vol. 38, no. 12, pp. 14439-14450, 2011.



Wu Deng

Wu Deng received his Ph.D. degree in computer science and technology from Dalian Maritime University in 2012. His research interests include artificial intelligence, the hybrid optimization algorithm, and combinatorial optimization.



Han Chen

Han Chen is master's degree candidate in computer technology in Dalian Jiaotong University from 2013. His main research interests include intelligent information processes.



He Li

He Li received his bachelor's degree in software engineering and English in Dalian Jiaotong University from 2010. His main research interests include the hybrid optimization algorithm.