International Journal of Smart Home
Vol. 9, No. 10, (2015), pp. 27-36
http://dx.doi.org/10.14257/ijsh.2015.9.10.04

A Two-Phase Hybrid Optimization Algorithm for Solving Complex Optimization Problems

Huiling Bao

Department of Communication and E-information, Shanghai Vocational College of Science & Technology, Shanghai 201800, China

Abstract

For solving traveling salesman problem (TSP), the ant colony optimization (ACO) algorithm and simulated annealing (SA) algorithm are used to propose a two-phase hybrid optimization (TPASHO) algorithm in this paper. In proposed TPASHO algorithm, the advantages of parallel, collaborative and positive feedback of the ACO algorithm are used to implement the global search in the current temperature. And adaptive adjustment threshold strategy is used to improve the space exploration and balance the local exploitation. When the calculation process of the ACO algorithm falls into the stagnation, the SA algorithm is used to get a local optimal solution. And the obtained best solution of the ACO algorithm is regarded as the initial solution of the SA algorithm, and then a fine search is realized in the neighborhood. Finally, the probabilistic jumping property of the SA algorithm is used to effectively avoid falling into local optimal solution. In order to verify the effectiveness and efficiency of the proposed TPASHO algorithm, some typical TSP is selected to test. The simulation results show that the proposed TPASHO algorithm can effectively obtain the global optimal solution and avoid the stagnation phenomena. And it has the better search precision and the faster convergence speed.

Keywords: *Ant colony optimization, simulated annealing, two-phase hybrid algorithm, adaptive selection, TSP*

1. Introduction

A great amount of problems in the industry, national defense, information and other fields can be converted to the combinatorial optimization problems. There are many intelligent algorithms [1-6], which are used to solve these combinatorial optimization problems, such as genetic algorithm(GA), Tableu search algorithm, simulated annealing(SA) algorithm, particle swarm optimization (PSO) algorithm, ant colony optimization(ACO) algorithm, neural networks, evolutionary algorithms, immune algorithm and so on. These intelligent algorithms have better optimized ability in the solving these combinatorial optimization problems. The traveling salesman problem (TSP) is a well-known combinatorial optimization problem and NP-complete problem, which finds a route covering each city once and only once with a minimum route distance by the salesman [7]. Due to the good ground of the TSP and its variants for solving optimization techniques, some researchers proposed many solving methods in various fields of artificial intelligence, biology, mathematics, physics, and operations research and so on.

Ant colony optimization (ACO) algorithm was introduced by Marco Dorigo in the early 1991[8]. It is a branch of artificial intelligence called swarm intelligence, which studies ‘‘the emergent collective intelligence of groups of simple agents”. When ants move, ants will leave a chemical pheromone trail on the ground. The indirect communication between ants via pheromone trails enables them to find shortest paths between their nest and food sources. So the ACO algorithm has increased interests for solving these optimization problems in recent years. But the

ISSN: 1975-4094 IJSH

ACO algorithm has some disadvantages of stagnation phenomenon, slow convergence, poor solution quality and easy falling into local optimal solution and so on in the actual engineering application. In allusion to the specific complex problems, a lot of researchers proposed a lot of improved ACO algorithm to solve the TSP. Liu Jenn-Long [9] proposed a rank-based ant colony optimization (ACO) method with a rank-based nonlinear selective pressure function and a modified Q-learning method to enhance the convergence characteristics of original ACO, which defines the probability of exploring a city to be visited by ants with a random proportional rule. Wu *et al.* [10] proposed a new method of the initialization of ants system based on studying the settings of parameters with experiments for different dimensions of TSP problems. Gao *et al.* [11] proposed a new algorithm which integrates ACO and AR to solve TSP problems. Wu *et al.* [12] proposed a population declining ant colony optimization (PDACO) algorithm to the traveling salesman problem (TSP) and multiuser detection. PDACO can enlarge searching range through increasing the initial population of the ant colony, and the population declines in successive iterations. Wang *et al.* [13] proposed an improved ACO algorithm in which some of ants can evolve by performing genetic operation, and the balance between intensification and diversification can be adjusted by numbers of ants which perform genetic operation. Gan *et al.* [14] proposed an ACO algorithm based on scout characteristic for solving the stagnation behavior and premature convergence problem of the basic ACO algorithm on TSP. Shuang *et al.* [15] proposed a hybrid PS-ACO algorithm, ACO algorithm modified by particle swarm optimization (PSO) algorithm. Mavrovounioti and Yang [16] proposed a hybridized ACO with local search (LS) (M-ACO), which is based on the population-based ACO (P-ACO) framework and an adaptive inver-over operator, to solve the DTSP. Li *et al.* [17] proposed a promising modification with changing index. You *et al.* [18] proposed a novel improved QACOT solving TSP and novel evolution mechanisms named measuring operator to improve the performance of QACOT. Elloumi *et al.* [19] proposed a novel approach by introducing a PSO, which is modified by the ACO algorithm to improve the performance. Saenphon *et al.* [20] proposed a new evolutionary optimization algorithm based on the actual manifold of objective function and fast opposite gradient search to improve the accuracy and speed of solution finding. Escario *et al.* [21] proposed a novel algorithm(ACE) belonging to the general Ant Colony Optimisation (ACO) framework. Mahi *et al.* [22] proposed a new hybrid method to optimize parameters that affect performance of the ACO algorithm using Particle Swarm Optimization (PSO). In addition, 3-Opt heuristic method is added to proposed method in order to improve local solutions.

These improved ACO algorithms can better solve the traveling salesman problems, but they exist some disadvantages. So the SA algorithm is introduced in to the ACO algorithm in order to propose a two-phase hybrid optimization algorithm, named TPASHO algorithm to solve the traveling salesman problem. The ACO algorithm is used to implement the global search, and adaptive adjustment threshold strategy is used to improve the space exploration and balance the local exploitation. In order to improve the search accuracy, the SA algorithm with memory function and the initial temperature of adaptive selection is introduced, the obtained best solution of the ACO algorithm is regarded as the initial solution of the SA algorithm, and then a fine search is realized in the neighborhood. Finally, the probabilistic jumping property of the SA algorithm is used to effectively avoid falling into local optimal solution.

2. Ant Colony Optimization and Simulated Annealing

2.1. Ant Colony Optimization (ACO) Algorithm

The ACO algorithm is an essential system based on agents that simulates the natural behavior of ants, including the mechanisms of cooperation and adaptation. It simulates the techniques employed by real ants to rapidly esTablelish the shortest route from a food source to their nest and vice versa without the use of visual information. The ACO algorithm consists of a number of iterations. In each iteration, many ants construct complete solutions by 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 the solution. The flow of the ACO algorithm is illustrated in Figure 1.

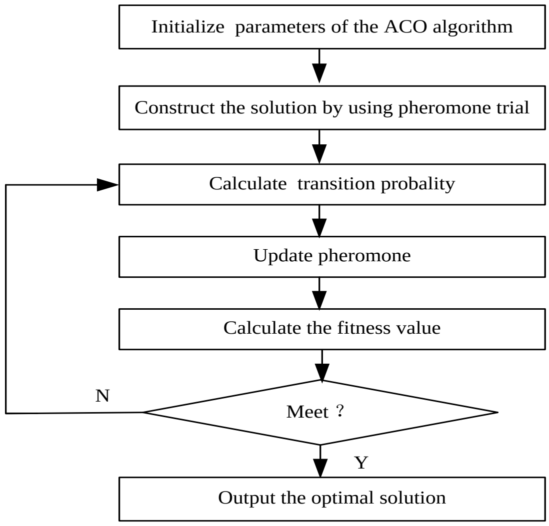


Figure 1. Flow of the ACO

Assume that there are n cities and m ants; the initial intensity of pheromone on each edge is set to a very small non-zero positive constant τ_0. In each cycle, each ant starts at one stochastic chosen city, then visits the other cities once and only once according to the transition rule based on the initial intensity of pheromone. The intensity of pheromone will be updated by the pheromone update rule. The pheromone update rule is shown as follows:

(1) The transition rule

In the route, the k^{th} ant starts from city r, the next city s is selected among the unvisited cities memorized in J_r^k according to the following expression:

$$s = \arg\max_{u \in J_r^k} [\tau_i(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}] \ \ if\ q \leq q_0 (Exploitati\ on) \tag{1}$$

To visit the next city s with the probability $p_k(r,s)$,

$$p_k(r,s) = \begin{cases} \dfrac{\tau(r,s)^{\alpha} \cdot \eta(r,s)^{\beta}}{\sum_{u \in J_r^k} \tau(r,u)^{\alpha} \cdot \eta(r,u)^{\beta}} & if\ s \in J_r^k \\ 0 & otherwise\ \ if\, q > q_0 (Bias\ Exploitati\ on) \end{cases} \quad (2)$$

(2) The pheromone update rule

In order to improve the solution, the pheromone trails must be updated. Trail updating includes local updating and global updating. The local trail updating formula is given by:

$$\tau(r,u) = (1-\rho)\tau(r,s) + \sum_{k=1}^{m} \Delta\tau_k(r,s) \quad (3)$$

In the formula (6), ρ ($0 < \rho < 1$) is the pheromone trial evaporating rate. $\Delta\tau_k(r,s)$ is the amount of pheromone trail added to the edge (r,s) by ant *k* between time *t* and t+Δt in the tour. It is given by:

$$\Delta\tau_k(r,s) = \begin{cases} \dfrac{Q}{L_k} & (\mathrm{r}, s) \in \pi_k \\ 0 & otherwise \end{cases} \quad (4)$$

Where Q is a constant parameter, L_k is the distance of the sequence π_k toured by ant *in*Δ*t*.

2.2. Simulated Annealing (SA) Algorithm

Simulated Annealing (SA) algorithm was developed by Kirkpatrick, followed by Aarts and Korstbased on the Metropolis algorithm dated from 1953. It is a computational stochastic technique to get near global optimum solutions. The SA algorithm is inspired from the thermodynamic process of annealing of molten metals to keep the lowest free energy state. When the molten metal is slowly cooled, it tends to solidify in a structure of minimum energy. This annealing process is mimicked by a search strategy. The core theory of SA algorithm is to allow the occasional worsening to move, so that they can eventually use the neighborhood to get the global minimum. The associated expression of the probability is given:

$$probability(p) = \exp\left(\frac{-\Delta E}{K_b T}\right) \quad (5)$$

In the expression (5), K_b is a constant, ΔE is the change in the energy value, T is the temperature.

For the optimization energy term, ΔE refers to the value of the function, T is a control parameter, which regulates the process of annealing. The acceptance criterion is popularly referred to as the Metropolis criterion. And the improving and deteriorating movement of the acceptance criterion has been proposed:

$$probability(p) = \frac{\exp(-\Delta E / T)}{1 + \exp(-\Delta E / T)} \quad (6)$$

The search strategy of the SA algorithm: any movement is accepted at the start. Then the temperature is gradually decreased. It shows that one becomes more and more selective in accepting new solution. In the end, the only improved movements are accepted in general. The temperature is systematically lowered by one problem-dependent schedule characterized.

3. The Two-Phase Hybrid Optimization (TPASHO) Algorithm

3.1. The IDEA of the TPASHO Algorithm

The hybrid algorithm is supposed to make best use of the advantages and bypass the disadvantages. The ACO algorithm can effectively solve complex optimization problem, it has strong robustness, excellent distributed computing mechanism, and it is easy to combine with other methods. But in solving complex problem, the ACO algorithm exists the shortcomings of poor search accuracy, and it is easy to fall into local optimum value. The SA algorithm is a new random search method, which a recently proposed method for solving large-scale combinatorial optimization problems. It has the simple description, flexible use, wide application, high efficiency and less constrained conditions. But it exists the shortcomings of the slow convergence speed, long running time, and its performance is sensitive to the initial values of the parameters. So the advantages of the ACO algorithm and the SA algorithm are used in order to propose a two-phase hybrid optimization (TPASHO) algorithm in this paper. The hybrid strategy of the TPASHO algorithm is: the advantages of parallel, collaborative and positive feedback of the ACO algorithm are used to implement the global search in the current temperature. And adaptive adjustment threshold strategy is used to improve the space exploration and balance the local exploitation. When the calculation process of the ACO algorithm falls into the stagnation, the SA algorithm is used to get a local optimal solution. And the obtained best solution of the ACO algorithm is regarded as the initial solution of the SA algorithm, and then a fine search is realized in the neighborhood. If the calculation result of the SA algorithm is better than the calculation result of the ACO algorithm, then the calculation result of the SA algorithm will be transformed into the pheromone of the ACO algorithm. Otherwise the original calculation of the ACO algorithm will continue. The TPASHO algorithm can effectively obtain the local search ability of the SA algorithm and the global search ability and convergence speed of the ACO algorithm by the multiple cycles.

3.2. The Flow of the TPASHO Algorithm

The flow of the two-phase hybrid optimization (TPASHO) algorithm based on combining the ACO algorithm and the SA algorithm is shown in Figure2.

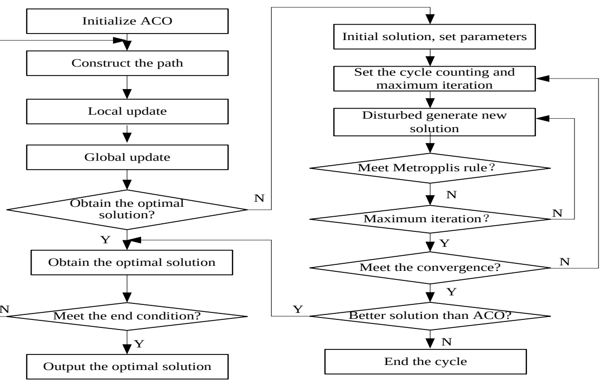


Figure 2. The Flow of the Two-Phase Hybrid Optimization (TPASHO) Algorithm

3.3. The Steps of the TPASHO Algorithm

The description of the two-phase hybrid optimization (TPASHO) algorithm is shown:

Step 1. Initialize the population and parameters, and execute the search calculation of algorithm.

Step 2. Construct the path and dynamically record the optimal value in the calculation of ACO algorithm.

Step 3. Execute the local update and global update.

Step 4. When the ACO algorithm has not obtained the better solution in the N iterations, the current optimal value is regarded as the initial value of the SA algorithm. Then the SA algorithm is used to search the solution.

Step 5. Set the parameters, the cycle counting and maximum iteration of the SA algorithm.

Step 6. Disturbed generate new solution.

Step 7. Determine to the maximum iteration. If it is, the next step is executed. Otherwise go to Step.6.

Step 8. Determine to the convergence. If it is, the next step is executed. Otherwise go to Step.5.

Step 9. If the obtained optimal path of the SA algorithm is better than the optimal path of the ACO algorithm, then the obtained optimal solution of the SA algorithm is translated into the pheromone matrix for giving the ACO algorithm. Otherwise the optimal path of the ACO algorithm the optimal path of the ACO algorithm is continued to use.

Step 10. Continue to execute the calculation of the ACO algorithm.

Step 11. Repeat Step 2-10 until the end condition is met.

Step 12. Output the optimal solution.

4. Experiment and Result Analysis

In order to test and verify the performance of the proposed TPASHO algorithm, some instances from TSPLIB are selected in this paper. According to the characteristics of TSPLIB, the distance between two cities is calculated by the Euclidian distance. In order to prove the effectiveness of the TPASHO algorithm, the SA and ACO algorithms are selected to compare with the TPASHO algorithm. The initial parameters of the three 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. These selected values of the parameters take on the optimal solution and the most reasonable running time of these algorithms to efficiently complete the problem solving. So the selected values of these parameters are shown in Table.1 for the SA, ACO and TPASHO algorithms.

Table 1. Parameters of the GA, ACO, GAACO and GASACO Algorithms

Parameters	SA	ACO	TPASHO
Population size (M)	N/A	30	30
Iteration time ($T_{\max}$)	1000	1000	1000
Initial temperature(T_0)	10	N/A	10
Temperature cooling ($T_{\max}$)	700	N/A	700
Pheromone factor(α)	N/A	1.0	1.0
Heuristic factor (β)	N/A	2.0	2.0
Evaporation coefficient ($\rho(t_0)$)	N/A	0.05	0.05
Pheromone amount (Q)	N/A	100	100

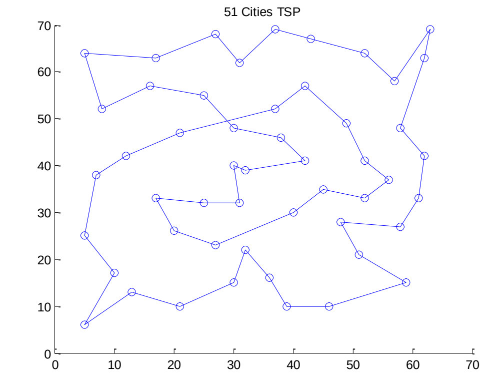
In the simulation experiments, each algorithm is carried out on Matlab platform and run on a 2.39GHz PC with 2GB of RAM running the Windows XP operating system. These algorithms are performed on 18 TSP benchmark instances with cities scale from 29 to 3038 for 20 times. The best value and average value are used to illustrate the optimization performance for all algorithms. The result of the simulated experiments is shown in Table. 2.

Table 2. The Results of the Simulated Experiments

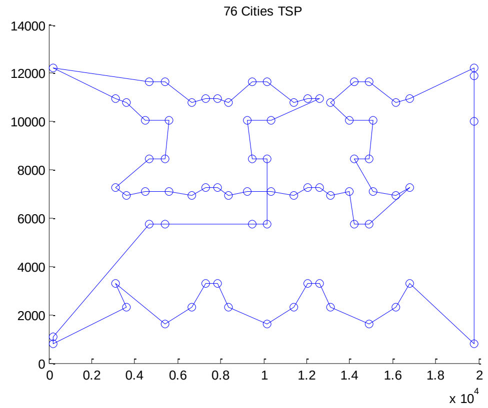
Instances	Opt.	SA		ACO		TPASHO	
		Best	Average	Best	Average	Best	Average
bayg29	1272	1377	1395	1303	1338	1287	1302
eil51	426	447	467	432	439	426	431
prl76	108159	112341	113523	110462	111037	108219	110031
rat99	1211	1317	1339	1266	1298	1218	1229
rad100	7910	8089	8138	7985	8001	7956	7994
lin105	14379	14535	14561	14465	14486	14413	14455
ch130	6110	6169	6201	6151	6177	6118	6146
u159	42080	42856	42962	42493	42514	42387	42431
kroA200	29368	29881	29993	29512	29542	29475	29496
pr264	49135	49798	49971	49417	49498	49329	49435
pr299	48191	48743	48898	48395	48483	48249	48323
rd400	15281	15775	15940	15464	15525	15333	15396
rat575	6773	6993	7093	6904	6992	6858	6911
u724	41910	42862	42966	42415	42516	42298	42392
rat783	8806	9109	9170	8997	9031	8949	8985
pr1002	259042	265702	266168	264151	265692	263283	264175
d1655	62128	64121	64516	63947	64341	63703	64180
pcb3038	137694	143389	144931	141107	141967	140598	141059

As can be seen from the Table.2, for the 17 TSP instances with the SA, ACO and TPASHO algorithms, the best value and the average value of the proposed TPASHO algorithm are the best for all 18 TSP instances in our experiment. For TSP instances eil51, the proposed TPASHO algorithm can search for the best known solutions 426. In addition, for TSP instances bayg29 , rat99 and ch130, the obtained best known solutions 1287, 1218 and 6118 are approaching to the best known solutions1272, 1211 and 6110. For larger scale instances, the experiment results show that the proposed TPASHO algorithm is better than the SA algorithm and ACO algorithm.

Figure3. Illustrates two best routes found by the TPASHO algorithm for TSP and their costs.



(1) eil51; Cost=426



(2) pr76; Cost=108219

Figure 3. The Best Routes Found by the TPASHO algorithm for TSP and their Costs

5. Conclusion

The ACO algorithm is an efficient and powerful optimization algorithm for solving complex function problems. The SA algorithm a computational stochastic technique to get near global optimum solutions, it has the simple description, flexible use, wide application, high efficiency and less constrained conditions. So the advantages of the ACO algorithm and the SA algorithm are used in order to propose a two-phase hybrid optimization (TPASHO) algorithm. The performance of the proposed TPASHO algorithm is evaluated on 18 TSP and is favorably compared with SA and ACO algorithms. The results demonstrate that the proposed TPASHO algorithm is more effective in obtaining better searching precision, convergence speed, stability, global convergence ability.

References

[1] E. Corchado and A. Abraham, “Hybrid intelligent algorithms and applications”, Information Sciences, vol. 180, no. 14, **(2010)**, pp. 2633-2634.
[2] P. Tarasewich and P. R. McMullen, “Swarm intelligence, Power in numbers”, Communications of the ACM, vol. 45, no. 8, **(2002)**, pp. 62-67.
[3] Z. Hua and F. Huang, “A variable-grouping based genetic algorithm for large-scale integer programming”, Information Sciences, vol. 176, no. 19, **(2005)**, pp. 2869-2885.
[4] I. Ellabib, P. Calamai and O. Basir, “Exchange strategies for multiple ant colony system”, Information Sciences, vol. 177, no. 5, **(2007)**, pp. 1248-1264.
[5] C. C. Lo and C. C. Hus, “Annealing framework with learning memory”, IEEE Transactions on System, Man, Cybernetics, Part A, vol. 28, no. 5, **(1998)**, pp. 1-13.

[6] T. A. Masutti and L. N. de Castro, "A self-organizing neural network using ideas from the immune system to solve the traveling salesman problem", Information Sciences, vol. 179, no. 10, **(2009)**, pp. 1454-1468.
[7] M. Dorigo and L. M. Gambardella, "Ant colony system, a cooperative learning approach to the traveling salesman problem", IEEE Transactions on Evolutionary Computation, vol. 1, no. 1, **(1997)**, pp. 53-66.
[8] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem", Bio systems, vol. 43, no. 2, **(1997)**, pp. 73-81.
[9] J. L. Liu, "Rank-based ant colony optimization applied to dynamic traveling salesman problems", Engineering Optimization, vol. 37, no. 8, **(2005)**, pp. 831-847.
[10] C. M. Wu, Z. Chen and M. Jiang, "Research on initialization of ants system and configuration of parameters for different TSP problems in ant algorithm", Acta Electronica Sinica, vol. 34, no. 8, **(2006)**, pp. 1530-1533.
[11] S. Gao, X. Y. Xia, L. Zhang and F. T. Zhuang, "Solving traveling salesman problem by ant colony optimization algorithm with association rule", Journal of Information and Computational Science, vol. 4, no. 3, **(2007)**, pp. 1011-1016.
[12] Z. L. Wu, N. Zhao, G. H. Ren and T. F. Quan, "Population declining ant colony optimization algorithm and its applications", Expert Systems with Applications, vol. 36, no. 3, **(2009)**, pp. 6276-6281.
[13] R. L. Wang, X. F. Zhou and K. Okazaki, "Ant colony optimization with genetic operation and its application to traveling salesman problem", IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, E92-A, no. 5, **(2009)**, pp. 1368-1372.
[14] R.W. Gan, Q. S. Guo, H.Y. Changu and Y. Yi, "Improved ant colony optimization algorithm for the traveling salesman problems", Journal of Systems Engineering and Electronics, vol. 21, no. 2, **(2010)**, pp. 329-333.
[15] B. Shuang, J. P. Chen and Z. B. Li, "Study on hybrid PS-ACO algorithm", Applied Intelligence, vol. 34, no. 1, **(2011)**, pp. 64-73.
[16] M. Mavrovouniotis and S. X. Yang, "A memetic ant colony optimization algorithm for the dynamic travelling salesman problem, soft Computing", vol.15, no. 7, **(2011)**, pp. 1405-1425.
[17] Y. C. Li, C. G. Ban, S. J. Zhou, X. Q. Li and R. Y. Li, "Novel ACO based on entropy and its application in analysis of structural reliability", Journal of Theoretical and Applied Information Technology, vol. 45, no. 1, **(2012)**, pp. 154-159.
[18] X. M. You, S. Liu and Y. M. Wang, "improved quantum ant colony algorithms, Advances in Information Sciences and Service Sciences, vol. 4, no. 22, **(2012)**, pp. 503-509.
[19] W. Elloumi, E. A. Haikal, A. Ajith and A. M. Alimi, "A comparative study of the improvement of performance using a PSO modified by ACO applied to TSP", Applied Soft Computing Journal, vol. 25, no. 1, **(2014)**, pp. 234-241.
[20] 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, **(2014)**, pp. 324-334.
[21] J. B. Escario, J. F. Jimenez and J. M. Giron-Sierra, "Ant colony extended, Experiments on the travelling salesman problem", Expert Systems with Applications, vol. 42, no. 1, **(2015)**, pp. 390-410.
[22] M. Mahi, Ö. K. Baykan and H. Kodaz, "A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem", Applied Soft Computing Journal, vol. 30, no. 5, **(2015)**, pp. 484-490.

Author

Huiling Bao, Lecture, received the Master degree in Control Theory and Control Engineering from Beijing Technology and Business University in 2003, Beijing, China. The main research fields: Information system and chaotic security communication, intelligent optimization algorithm.