ISSN 1009-3095 *Journal of Zhejiang University SCIENCE* V.4, No.1, P.40-46, Jan.-Feb., 2003
http://www.periodicals.com.cn; http://www.zju.edu.cn/English
http://www.zjupress.com; http://lib.zju.edu.cn/eindex.htm; jzu_s@mail.hz.zj.cn

An adaptive ant colony system algorithm for continuous-space optimization problems*

Li Yan-jun(李艳君)†, Wu Tie-jun(吴铁军)
(*Institute of Intelligent Systems and Decision Making, Zhejiang University, Hangzhou 310027, China*)
† E-mail: yjlee@iipc.zju.edu.cn
Received Dec. 29, 2001; revision accepted June 16, 2002

Abstract: Ant colony algorithms comprise a novel category of evolutionary computation methods for optimization problems, especially for sequencing-type combinatorial optimization problems. An adaptive ant colony algorithm is proposed in this paper to tackle continuous-space optimization problems, using a new objective-function-based heuristic pheromone assignment approach for pheromone update to filtrate solution candidates. Global optimal solutions can be reached more rapidly by self-adjusting the path searching behaviors of the ants according to objective values. The performance of the proposed algorithm is compared with a basic ant colony algorithm and a Square Quadratic Programming approach in solving two benchmark problems with multiple extremes. The results indicated that the efficiency and reliability of the proposed algorithm were greatly improved.

Key words: Ant colony algorithm, Continuous-space optimization, Pheromone update strategy
Document code: A **CLC number:** TP202

INTRODUCTION

Ant colony algorithms (Hertz, et al., 2000), investigated systematically at first in Dorigo's Ph.D. dissertation (1992) as the imitation of the food-seeking behavior in ant societies, have attracted the great attention of researchers in comprehensive fields of system optimization with high complexity, e.g., communication network routing, multi-robot job assignment, dynamical data mining and graph creation and partitioning (Dorigo et al., 2000). Because of their biological background, the basic ant colony algorithms developed by Dorigo and extended by others are designed to solve complicated combinatorial optimization problems (eg., TSP, and QAP) and little has been done for the search in continuous-spaces (Bilchev et al., 1995; Zhang et al., 2000; Dorigo et al., 2000). Some aspects such as pheromone assignment and update methods, route searching approaches and optimal solution reservation strategies in the basic ant colony algorithms could be improved (Song et al., 1999). Empiricism and intuition remain in those algorithms due to their short development history and lack of rigorous theoretical validation and practical implementation. As a result, the basic ant colony algorithms have low efficiency, poor convergence, and divergence of searching results (Stützle et al., 2000; Dorigo et al., 1996).

To break through the limitations of the basic ant colony algorithms described above, an adaptive ant colony algorithm is proposed here to solve the optimization problems in continuous-spaces, benefiting from the encoding techniques and elite reservation strategies used in genetic algorithms and local region search featured in hill-climbing approaches. The adaptive ant colony algorithm adopts a novel objective-value-based heuristic pheromone assignment mechanism. This algorithm can adaptively adjust the route searching behavior of ants according to objective values to make more refined search attempts within hopeful regions, and explore wider space by keeping diversity in route selection, and thus find global solutions rapidly.

BASIC ANT COLONY ALGORITHM

To present the algorithm developed in this paper more clearly, the graph-based ant system

* Project (No.9845-005) supported by National High-Tech. Research & Development Plan, China

proposed by Gutjahr (2000) is briefly described first in this section.

1. An oriented graph is defined as $C = (V, S)$, where V is the set of nodes in the graph C, and S the set of oriented arcs among those nodes. A path, connecting an initiative node and a terminal node through a series of interim nodes by oriented arcs, is denoted by ω. A function ϕ maps the path ω in the graph C to a feasible solution of an optimization problem, under the constraint that no path loop exists in this graph.

2. The set of n ants in an ant colony is denoted by $A = \{A_1, \cdots, A_n\}$. In each ant searching period, an ant randomly chooses a path ω in the graph C according to a predetermined path selection possibility. In this way, a path seeking movement by an ant corresponds to the search for a feasible solution of a shortest route problem. A searching period ends up in the algorithm when all the n ants finish their path seeking respectively. The path selection possibility $P_{i,j}(t)$ for a search from node i to node j (denoted by the path segment (i,j)) in the searching period t is defined by

$$p_{i,j}(t) = \frac{\varphi_{i,j}^{a}(t) \cdot d_{i,j}^{b}}{\sum_{(i,k)\in S,\, k \notin U} \varphi_{i,k}^{a}(t) \cdot d_{i,k}^{b}}, \tag{1}$$

where U is the partial route that had been searched by ants in the period t; $\varphi_{i,j}(t)$ is the density of pheromone accumulated at the path segment (i,j) by ants in the period t, implying the ant's searching tendency in the next period; $d_{i,j}$ is the cost of searching on path segment (i, j), usually representing distance or price, etc; and a, b are both positive real numbers, called pheromone index and cost index, respectively.

3. The pheromone $\varphi_{i,j}(t)$ on the path segment (i,j) will be updated according to the behavior of the ant colony system at the end of each searching period in such a way that

$$\varphi_{i,j}(t+1) = \lambda\varphi_{i,j}(t) + (1-\lambda)\Delta\varphi_{i,j}(t), \tag{2}$$

where λ is called the evaporation factor, and $0 < \lambda < 1$; $\Delta\varphi_{i,j}$ is a pheromone increment correlated with the objective values of feasible solutions corresponding to all the paths passing through the path segment (i, j) as a non-increasing function of objective values, e.g.,

$$\begin{aligned} \Delta\varphi_{i,j} &= \sum_{k=1}^{n}\Delta\varphi_{i,j}^{(k)} \Big/ \sum_{(i,j)\in S}\sum_{k=1}^{n}\Delta\varphi_{i,j}^{(k)}, \\ \Delta\varphi_{i,j}^{(k)} &= \begin{cases} \psi(f_k), & \text{if } (i,j) \text{ is passed by ant } k; \\ 0, & \text{otherwise}; \end{cases} \end{aligned} \tag{3}$$

where f_k is the objective value corresponding to the searching path of the ant A_k; and $\Psi(\cdot)$ is a non-increasing function. The pheromone increment vanishes for any path that has not been searched yet in the current period.

4. The algorithm ends when the number of searching periods reaches a predetermined value. This is a commonly used criterion because of its random searching characteristics.

A rigorous proof for the convergence of the graph-based ant colony algorithm was given (Gutjahr, 2000) under several conditions. For the problems to be solved in this paper, the shortcomings of the algorithm include:

1) How to map continuous space optimization problems to the graph-based ant colony system;

2) How to update more efficiently the pheromone on the paths searched by ants to prevent the algorithm from making a large number of invalid search efforts, due to the fact that, according to Eq. (3), all the searching paths passing through the path segment (i,j) contribute pheromone increments to it;

3) How to assign pheromone to each segment in a path to emphasize the significance of certain "best" path segments in order to speed up the convergence of the algorithm. Also according to Eq. (3), a uniform assignment strategy is used in the original algorithm, with no consideration for the influence of searching scale on the efficiency of the algorithm.

ADAPTIVE ANT COLONY ALGORITHM FOR CONTINUOUS-SPACE OPTIMIZATION PROBLEMS

1. Mapping continuous-space optimization problems to an oriented-graph searching problem

The continuous-space optimization problem to be solved in this paper is as follows:

$$\begin{cases} \min\ J = f(x) \\ x_{\min} \leqslant x \leqslant x_{\max};\ \ x \in R^n \end{cases} \tag{4}$$

The complexity of the problem expressed in Eq. (4) lies in the objective function $f(x)$ that may not be formulated analytically or even mathematically. As the first step to solve this problem by using ant colony search, we express the solution candidate x with N-length binary string:

$$x \Leftrightarrow \{b_N b_{N-1} \cdots b_1\}, \tag{5}$$

where $b_j \in \{0, 1\}$ for $j = 1, 2, \cdots, N$; b_1 is the lowest bit and b_N is the highest bit in the string. An oriented graph $C = (V, S)$ is defined with the set of nodes

$$V = \{v_s, v_N^0, v_{N-1}^0, \cdots, v_1^0, v_N^1, v_{N-1}^1, \cdots, v_1^1\}, \tag{6}$$

and the set of oriented arcs

$$S = \{(v_N^0, v_{N-1}^0), \cdots, (v_j^0, v_{j-1}^0), (v_j^0, v_{j-1}^1), (v_j^1, v_{j-1}^0), (v_j^1, v_{j-1}^1), \cdots, (v_2^1, v_1^1)\} \tag{7}$$

where the node v_s is the unique initiative node; the nodes v_j^0 and v_j^1 for each j represent two states of the bit b_j, i.e., 0 and 1, respectively. At the nodes v_j^0 and v_j^1 for $j = 2, 3, \cdots, N$, there exist only the arcs that point to the nodes v_{j-1}^0 and v_{j-1}^1.

The ant A_i begins its search from the node v_s, along the path segment composed of the arcs through N nodes, to form a path ω_i consisting of the nodes sequence $\{v_N^{i_N}, v_{N-1}^{i_{N-1}}, \cdots, v_1^{i_1}\}$ for $i_1, \cdots, i_N \in \{0, 1\}$, corresponding to the binary string $\{b_N^i b_{N-1}^i \cdots b_1^i\}$. The sequence can be decoded to a solution candidate x_i:

$$\Phi(\omega_i) = \{b_N^i b_{N-1}^i \cdots b_1^i\} \Leftrightarrow x_i = \frac{X_i}{2^N - 1} \cdot (x_{\max} - x_{\min}) + x_{\min}, \tag{8}$$

where X_i is the binary number of the string $\{b_N^i b_{N-1}^i \cdots b_1^i\}$, to guarantee all the searching paths satisfy the constraint $x_{\min} \leqslant x \leqslant x_{\max}$ under the mapping of Eq. (8).

2. Pheromone increment assignment strategy based on bit position

Based on the uniform pheromone assignment strategy in the basic ant colony algorithm, $\psi(f_k)$ is independent of the order of the segments in the searching path. However, according to the proposed mapping from continuous-space optimization problems to a graph searching process, the preceding nodes in a searching path correspond to the upper bits of an encoded solution candidate. The variation of the amount of pheromone on the segments among those nodes may result in a great change in the value of the solution candidate. The pheromone on the segments among the posterior nodes in the path will produce reverse effects. This uniform pheromone assignment strategy induces the same probability in the choice of different searching step lengths in the next searching period, when the new search is made on the basis of those paths.

The searching technique based on the uniform pheromone assignment strategy does not match the conclusion derived from the analysis on the topological structure of solution spaces (Zhang et al., 2000). The reasonable searching strategy should be that, on the one hand, for better solution candidates the search in the next period is apt to be conducted in a smaller region of the solution space in order to gain better convergence in the probabilistic sense (Preux et al. 1999); on the other hand, for worse solution candidates, in the next period a bigger search step should be set for global optimality.

We present a bit-position-based pheromone increment assignment strategy to tackle the problem mentioned above.

Suppose that, in the t'th searching period, the node i of the path segment (i, j) in the searching path $W_s(t)$ corresponds to the k'th bit of the binary-encoded solution candidate x_s, and $f_s(t)$ is the objective value corresponding to the path $W_s(t)$. According to the principle of smaller searching steps for better solutions, a new method for the pheromone increment calculation is defined as

$$\Delta\varphi_{i,j}(f_s(t), k) = \frac{1}{1 + e^{\beta \cdot k \cdot f_s(t) \cdot (f_s(t) - (f_{\min}(t) + \delta))}} \tag{9}$$

where $f_{\min}(t)$ is the minimum of the objective values in the last t searching periods, $\beta, \delta > 0$. The function on the right side of Eq. (9) is shown in Fig. 1, and has the following characteristics:

if $f_1 < f_2$, then $\Delta\varphi_{i,j}(f_1, k) > \Delta\varphi_{i,j}(f_2, k)$,

if $k_1 < k_2$ and $f_s \leqslant f_{\min} + \delta$, then $\Delta\varphi_{i,j}(f_s, k_1) < \Delta\varphi_{i,j}(f_s, k_2)$,

if $k_1 < k_2$ and $f_s > f_{\min} + \delta$, then $\Delta\varphi_{i,j}(f_s, k_1) > \Delta\varphi_{i,j}(f_s, k_2)$,

Combining Eq. (9) and Eq. (1), it can be seen that, at the end of the searching period t, the worse the solution candidate, or the lower the encoding bit position, the smaller the probability is in which the corresponding node will be searched. In this way in the searching period $t+1$, it is more likely to make a small-scale step search based on the original solution candidate, to avoid losing effective information provided in the previous searching periods.

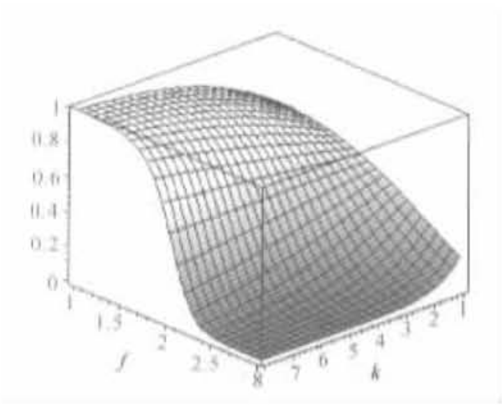

Fig. 1 Pheromone increment assignment function

SIMULATION STUDY

To validate the algorithm proposed in this paper, the adaptive ant colony algorithm (AACA) was used in the simulation study to solve two benchmark problems. The basic ant colony algorithm (BACA) as well as a conventional Sequential Quadratic Programming (SQP) method were also used to solve the same problems for comparison.

Test 1: The Combinatorial Exponential Sinusoid (CESIN) problem

$$\min J_1(x) = 5e^{-0.5x}\sin 30x + e^{0.2x}\sin 20x + 6, \quad x \in [0, 8]. \tag{10}$$

The property of the objective function J_1 is shown in Fig. 2. Obviously, It has a large number of extremes among which the global minimum is $J_1^* = 1.2573$ that is very sensitive to the variation of the argument x.

Test 2. The Six-hump Camel Back (SCB) problem (Michalewicz, 1996):

$$\min J_2(x_1, x_2) = (4 - 2.1x_1^2 + \frac{1}{3}x_1^4)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2; \quad x_1 \in [-2, 2]; \quad x_2 \in [-1, 1]. \tag{11}$$

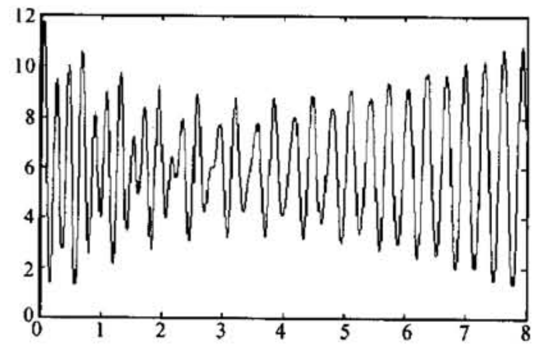

Fig. 2 Objective function of the CESIN problem

The characteristic of the objective function J_2 can be seen in Fig. 3. There are total six extremes within the predefined feasible region, among which there is a global minimum $J_2^* = -1.0316$ at two different points $(x_1, x_2) = (-0.0898, 0.7126)$ and $(-0.0898, -0.7126)$.

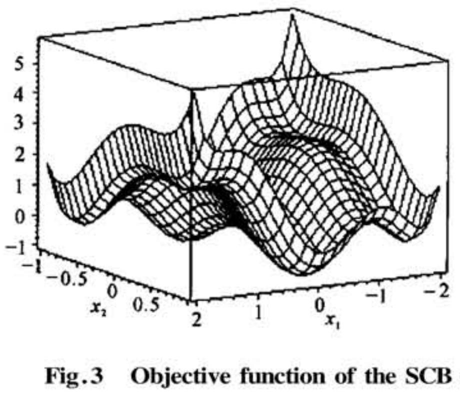

Fig. 3 Objective function of the SCB problem

For comparison, the parameters in the two algorithms were set the values listed in Table 1.

In view of the probabilistic characteristics of both AACA and BACA, as well as the fact that the effectiveness of SQP strongly depends upon the choice of the initial points from which the algorithm starts its search, comparison of their performance is done in a statistical way. The statistical indexes taken into consideration for each tested algorithm include:

Table 1 Parameter settings of AACA and BACA

Test	Parameters					
	Number of searching periods	Size of ant population	Encoding length	Evaporation factor	Pheromone index	Cost index
Test 1	20	10	8	0.2	0.1	0
Test 2	40	20	8	0.5	0.8	0

1. The minimal objective value found by the tested algorithm in N rounds of computations is defined by

$$J^* = \min_{k \in \{1,2,\cdots,N\}} \{J^{(k)}\};$$

where $J^{(k)}$ denotes the best objective value obtained in the k-th round of problem solving; this index is a direct measure for the ability of the tested algorithm to search for a global optimal solution;

2. The mean value of the best objective values found in the N rounds of computations is defined by

$$\bar{J} = \frac{1}{N}\sum_{k=1}^{N} J^{(k)}$$

which can be used to estimate the repetitiveness of the tested algorithm in the search for a global optimal solution;

3. The relative error of the mean value $\bar{J}$ of the minimal objective value J^* among the best objective values found in the N rounds of computations is defined by

$$\varepsilon = \left| \frac{\bar{J} - J^*}{J^*} \right|$$

which can also be used as an alternative index to estimate the repetitiveness of the tested algorithm in searching for an optimal solution of a tested problem, or, furthermore, to compare the ability of the tested algorithm in searching for corresponding optimal solutions of different problems;

4. The standard deviation of the objective values found in N computations of the mean value $\bar{J}$ is defined by

$$\sigma = \left(\frac{1}{N}\sum_{k=1}^{N}(J^{(k)} - \bar{J})^2\right)^{\frac{1}{2}},$$

which can be used to evaluate the divergence of the solutions found by the tested algorithm, or, in other words, the stability of the algorithm.

One thousand independent solutions of the CESIN problem and the SCB problem were tested by AACA, BACA and SQP, respectively, in such a way that in each round of computations a choice of initial path selection probability (for both AACA and BACA) or initial start points (for SQP) was made at random. The corresponding statistical results are listed in Table 2 and Table 3.

Table 2 Performance comparison for 1000 rounds of CESIN problem solving

Algorithms	J^*	$\bar{J}$	ε	σ
AACA	1.3652	1.4403	0.0550	0.0061
BACA	1.3652	1.5311	0.1215	0.0104
SQP	1.2573	4.3420	2.4534	1.9849

Notes: The best objective value found by AACA and BACA is worse than that by SQP due to the limitation of the encoding length.

Table 3 Performance comparison for 1000 rounds of SCB problem solving

Algorithms	J^*	$\bar{J}$	ε	σ
AACA	−1.0315	−1.0299	0.0016	0.0034
BACA	−1.0314	−0.8718	0.1547	0.1455
SQP	−1.0316	−0.7341	0.2883	0.4967

Statistical results based on the 1000 rounds of computations indicated that, on the one hand, for the continuous-space optimization problems with multiple local extremes, both AACA and BACA were greatly superior to the conventional SQP method using gradient or curvature information to guide its search; on the other hand, the optimal-solution-searching ability and algorithm stability of AACA were better than those of BACA. For example, in comparison with BACA, the relative error of the mean value of the best objective value found in 1000 rounds of computations conducted by AACA was decreased by 54.73% and 98.96% respectively in two tests, and the corresponding standard deviation of the objective values was decreased by 41.34% and 97.66%. These results indicated that, the efficiency and reliability of the adaptive ant colony algorithm proposed in this paper have been greatly improved.

Fig. 4 shows the trend of searching performance of AACA, represented by the best objective values obtained in all searching periods, in a typical round of computations of Test 2. The distribution of the solution candidates in the solution space in the 1st, 8th, and 40th periods in Test 2. is shown in Fig. 5. It can be seen that, the solution candidates at the beginning of the algorithm were randomly distributed in the solution space; with the elapse of searching time, however, more and more solutions influenced by the pheromone got closer to the current better solutions.

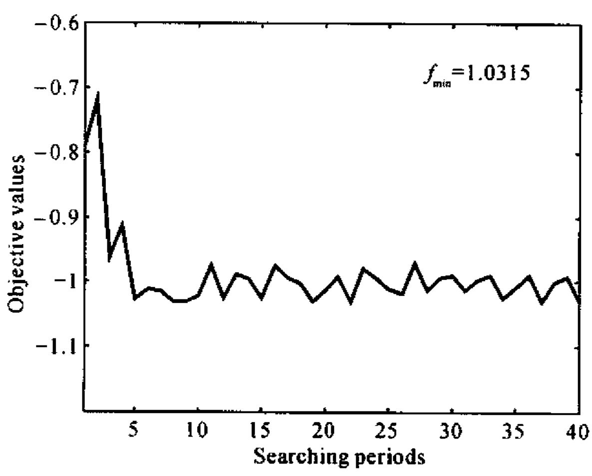

Fig. 4 Performance of AACA in Test 2
(The best objective values to n solution candidates in all searching periods)

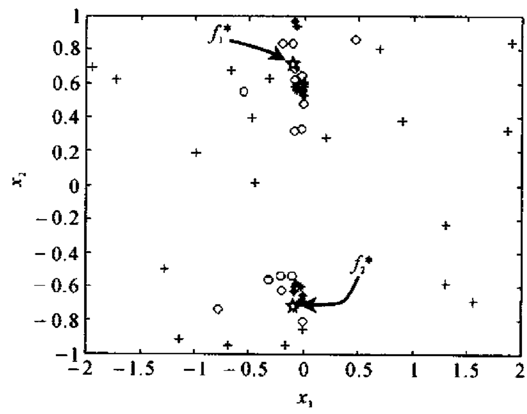

Fig. 5 Solution candidates in different searching periods of AACA in Test 2
(The symbols of plus. circle and star represent the solution candidates in the 1st. 8th. and 40th period, respectively)

A practical problem of complicated hybrid production scheduling in flexible processing industries was solved via a revised version of nested AACA (Li et al., 2002), demonstrating its strong capability for solving hard optimization problems with coupled continuous-time and discrete-event variables.

CONCLUSIONS

In this paper we propose an adaptive ant colony algorithm to tackle continuous-space optimization problems. This approach uses a new objective-function-based heuristic pheromone assignment method for pheromone update, ensuring that the assignment of pheromone in the path segments is positively proportional to the optimality of the solutions. Besides, this approach uses the bit encoding information for pheromone update to avoid inefficient searches.

The applicable fields of the proposed algorithm include, but are not limited to:

1. problems with multiple local extremes due to the multiple-points-random-searching characteristics of the proposed approach;

2. problems without structural objective expressions such as equations or formulas because no gradient or curvature information are needed in the proposed algorithm;

3. problems with hybrid data structures such as the combination of continuous-time variables and discrete-event variables via the introduction of encoding mechanism;

4. problems with large scale owing to the parallelity of the ant colony algorithms in nature.

References

Bilchev, G. A. and Parmee, I. C., 1995. The ant colony metaphor for searching continuous design spaces. *Lec-*

ture Notes in Computer Science, **993**: 25 – 39.

Dorigo, M., 1992. Optimization, learning, and natural algorithms. Ph. D. Thesis, Dip Elettronica, Politecnico di Milano, Italy.

Dorigo, M., Bonabeau, E. and Theraulaz, C., 2000. Ant algorithms and stigmergy. *Future Generation Computer Systems*, **16**: 851 – 871.

Dorigo, M., Maniezzo, V. and Colorni, A., 1996. Ant system: optimization by a colony of cooperating agents. *IEEE Trans. On Systems, Man and Cybernetics*, **26** (1): 28 – 41.

Dorigo, M., Caro D. C. and Stützle T., 2000. Ant algorithms. *Future Generation Computer Systems*, **16**: p. V – vii.

Gutjahr, W. J., 2000. A graph-based ant system and its convergence. *Future Generation Computer System*, **16**: 837 – 888.

Hertz, A. and Kobler, D., 2000. A framework for the description of evolutionary algorithms. *European Journal of Operational Research*, **126**: 1 – 12.

Michalewicz, Z., 1996. Genetic algorithms + date structures = evolution programs. Springer -Verlag Berlin Heidelberg.

Li, Y., Wu, T.-J., 2002. A nested ant colony algorithm for hybrid production scheduling. Proceedings of the American Control Conference. *Anchorage*, AK: 1123 – 1128.

Preux, Ph. and Talbi, E.-G., 1999. Towards hybrid evolutionary algorithms. *Intl. Trans. in Operational Research*, **6**: 557 – 570.

Song, Y. H., Chou, C. S. and Stonham, T. J., 1999. Combined heat and power economic dispatch by improved ant colony search algorithm. *Electric Power Systems Research*, **52**: 115 – 121.

Stützle, T. and Hoos, H. H., 2000. Max-Min ant system. *Future Generation Computer Systems*, **16**: 889 – 914.

Zhang, J., Gao, Q. and Xu, X., 2000. A self-adaptive ant colony algorithm. *Control theory and applications*, **17**(1): 1 – 8.