Applied Mathematics & Information Sciences
An International Journal

http://dx.doi.org/10.12785/amis/072L47

Cultural Ant Algorithm for Continuous Optimization Problems

Jianmin Xu[1], *Minjie Zhang*[1] *and Yanguang Cai*[2]

[1]School of Civil Engineering and Transportation, South China University of Technology, 510641 Guangzhou, P. R. China
[2]Faculty of Automation, Guangdong University of Technology, 510006 Guangzhou, P. R. China

Received: 20 Oct. 2012, Revised: 3 Feb. 2013, Accepted: 6 Feb. 2013
Published online: 1 Jun. 2013

Abstract: In order to overcome prematurity of ant colony algorithm, the conception of belief space originated in cultural algorithm is introduced, and a new cultural ant algorithm is proposed for continuous optimization problems. Firstly, the coding scheme for ant colony algorithm to solve continuous optimization problems is discussed. Then belief space is brought in, and designed as the form of two parts: individual belief space and population belief space. The former is used to conduct individuals' deep search for better solutions, and the other to help worse individuals drop their current bad solution space for broad search. The update rules of both population space and belief space are given subsequently. Eight common standard functions are used to test the new algorithm, which is compared with four other algorithms at the same time. The results show effectiveness and superiority of the new algorithm. Finally the effect of the parameter used in the algorithm is discussed, and so does the both two belief space.

Keywords: Global optimization, continuous problems, intelligent algorithm, cultural ant algorithm, belief space

1 Introduction

Ant colony algorithm firstly proposed by M. Dorigo in 1992, imitates the process of ants group foraging [1]. For the advantages of not depending on mathematics description of definite problems, excellent capacity on global optimization, better performance on reliability than early genetic algorithm and annealing simulation algorithm, little workload, and easy to realization, it has been paid more attentions to solve discrete problems, such as combination optimization and modification consistency of judgment matrix [2,3,4].

But ant colony algorithm still has defects. First of all, it was original designed for the discrete problems, so it cannot be used to solve continuous optimization problems directly. Secondly, its convergence mechanism is based on positive feedback, which may not only help the algorithm accelerate convergence, but also make the algorithm being prone to be premature. Many methods were introduced to improve the ant algorithm seen in other research, which were mostly concentrated on the update of pheromone and the generation of initial solutions [5,6]. These approaches promote the performance of the algorithm on overcoming premature to some extent, but not thoroughly; they have finite effect on improving search ability for better solutions either. The cultural algorithm is a particular class of evolutionary algorithm that uses domain knowledge extracted from solutions during evolutionary process to improve the performance of the search engine (i.e. the evolutionary algorithm) adopted [7,8]. It has natural complementarity with other intelligent algorithms. In this paper, cultural algorithm and ant colony algorithm are combined to construct a new algorithm called cultural ant algorithm (abb. CACO) to against these defects.

The following is the organization of the paper. In Section 2, existing coding schemes for ant colony algorithm for continuous optimization problems are discussed firstly, and then valid coding scheme that discretizes continuous space to translate continuous problems to discrete problems is chosen for the new algorithm. Then the conception of cultural: belief space is design as the form of two parts: individual belief space and population belief space. And then update rules of solutions and both two spaces are designed in Section 2.3 and 2.4. In Section 2.5, the steps of CACO are given. Numerical studies to test CACO are laid in Section 3. The influence analysis of the parameter and cultural operators

* Corresponding author e-mail: wdyhdz@126.com

are given in Section 3.2. Section 4 concludes the whole paper and describes the future research direction.

2 Algorithm Design

Cultural algorithm and ant colony algorithm are combined to construct CACO, which effectively use knowledge guidance mechanism against premature.

Ant colony algorithm was original designed for discrete problems, such as TSP [9]. It cannot be used in continuous scenarios directly. There are two methods for solving these scenarios: (1) continuous space discretization, thus continuous problems are translated to discrete problems; (2) combine with other continuous algorithm, and use the real coding directly. If the latter method is chosen, because of the unlimitedness of real values, the advantage of positive feedback will not be given full play. So the former is chosen in the paper.

2.1 Coding

If the continuous solution space is discretized directly as a structure like TSP, ants have n choices at each variable for a value. For ensuring precision of the algorithm,n may be a big number exceed the amount of ants group. That means many paths will not have their pheromone incremental after one iteration. Because of positive feedback mechanism, the algorithm is easy to be premature.

In order to solve this problem, the binary coding scheme is proposed. The structure for regular solutions search is shown in Fig.2.1.

Assume the space of variable i is $[a_i, b_i]$. Discretize this space to m small areas. The length of each area is $(b_i - a_i)/m$. The middle value of each area is used to present each small area. The binary bits to present each variable is n, it satisfies $2n-1 \leqslant m \leqslant 2n$. Ants start from S, and seek for their valid paths to E, depending on pheromone concentration among adjacent points, and then regular solutions is gotten.

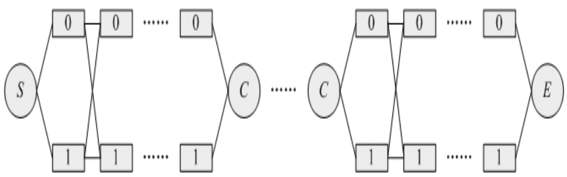


Figure 2.1: The structure for solutions search

2.2 Belief Space

The core idea of belief space is to describe and update the knowledge found in past iterations. In this paper the space of variables in which potential better solutions lay is used as the belief space. The belief space includes the individual belief space and the population belief space.

2.2.1 Design of Belief Space

(1) Individual Belief Space

Each individual has its belief space in evolution process. By evaluating solutions at current generate-on, some useful information for the next search will be acquired. This information is the belief space of individuals. For derivative problems, the partial derivative of solutions can support this information, so the belief space of each individual is generated by calculating the partial derivative of its solution.

Take minimal optimization problems as examples. Assume the solution of individual i at current generation gen (gen=1,2,...) is:

$$sln_i^{gen} = \{x_{i1}, x_{i2}, ..., x_{in}\}$$

Its partial derivative is:

$$\Delta s_i^{gen} = \{\Delta x_{i1}, \Delta x_{i2}, ..., \Delta x_{in}\}$$

Obviously, the search diretion of j-th variable at (gen+1) is $-\Delta x_{ij}$. Then the belief space of individual i at j-th variable is defined as:

$$blf_{ij}^{gen} = \begin{cases} (x_{ij}, x_{ij} - rand * \Delta x_{ij}), & \text{if } \Delta x_{ij} < 0 \\ (x_{ij} - rand * \Delta x_{ij}, x_{ij}), & \text{if } \Delta x_{ij} > 0 \end{cases}$$

Where, $rand$ is a random value in (0, 1).

(2) Population Belief Space

The population belief space will be gained from statistics in search process. It records the distribute-on of excellent individuals in global solution space, extracts potential good area, and then sequentially executes dynamic division at the solution space, for inducing global search. The topological knowledge [10] is used as the representation of the population belief space. If some current individuals are lack of competitiveness, they will be guided by the population belief space.

Take 2d variables optimization as an example. Assume the solution space is $[a_0, b_0]$ for both two variables, and the degree of division is set to 2. At the first search process, the solution space is divided into four subregions, all of which have same size. The superiority of these subregions is described by regional attribute, which is calculated by the current average fitness of individuals in the subregion, the fitness of the best individual in the subregion, and the average fitness of current group. Remember bS^k as the best individual in subregion k, and $ftn(bS^k)$ as its fitness value; $afit_{gen}^k$ as current average fitness of individuals in subregion k, then regional attribute of subregion k is :

$$atb_{gen+1}^k = (\frac{afit_{gen}^k}{aveFit_{gen}})^\alpha \cdot (\frac{ftn(bS^k)}{aveFit_{gen}})^\beta \tag{1}$$

Then, the probability of using the subregion k to induce individuals that are lack of competitiveness for optimization is:

$$p^k_{gen+1} = \frac{atb^k_{gen+1}}{\sum_{j=1}^{T_{gen+1}} atb^j_{gen+1}} \tag{2}$$

In which, T_{gen+1} is the total number of subregions at generation $gen+1$. That is, new individuals which at generation gen are lack of competitiveness will be generated in the subregion k, which is gained by a process of roulette.

2.2.2 Update of the Belief Space

(1) Update of individual belief space

Assume current generation is gen. The local best solution of individual i is sln^{local}_i, and its current solution is sln^{gen}_i. The fitness of sln^{gen}_i is $ftn(sln^{gen}_i)$; $mCnt$, which describes the maximal continuous times that CACO has not found better solution after the current best history solution was found, and cnt is the continuous times. If individual i satisfies:

$ftn(sln^{gen}_i) > ftn(sln^{local}_i)$

Or satisfies:

$ftn(sln^{gen}_i) < ftn(sln^{local}_i)$, $ftn(sln^{gen}_i) > aveFit_{gen}$ and at the same time $cnt < mCnt$.

Then, the individual i is regarded as an excellent individual at current generation.

Excellent individuals update their belief space with their current belief space and their local belief space. The update principle is to contract the length of space as much as possible, helping to realize more effective search.

Take individual i as an example. Its local best belief space at j-th variable is:

$$blf^{local}_{ij} = (x^{local}_{low}, x^{local}_{upp})$$

And its current solution at j-th variable is x_{ij}.

If $\Delta x_{ij} < 0, \Delta x^{local}_{low} < 0, blf^{gen}_{ij} = (x_{ij}, x^{upp}_{ij})$, and $x^{local}_{upp} < x^{upp}_{ij}$. There are 2 situations:

1) $\Delta x^{local}_{upp} < 0$. Update the local best belief space to blf^{gen}_{ij}. That means $blf^{local}_{ij} = blf^{gen}_{ij}$;

2) $\Delta x^{local}_{upp} > 0$. Update the local best belief space as $(x_{ij}, x^{local}_{upp})$. That means $blf^{local}_{ij} = (x_{ij}, x^{local}_{upp})$.

When $x^{local}_{upp} > x^{upp}_{ij}$, there are also 2 situations:

1) $\Delta x^{upp}_{ij} > 0$. Update the local best belief space to blf^{gen}_{ij}. That means $blf^{local}_{ij} = blf^{gen}_{ij}$;

2) $\Delta x^{upp}_{ij} < 0$. Update the local best belief space to blf^{gen}_{ij}. The update method at other situations is similar to the above.

(2) Update of the population belief space

There are two situations to update the population belief space:

1) Subregions refining. If current average fitness of individuals in subregion k is greater than current average fitness of the group, or though the former is not satisfied, the fitness of the best individual in the subregion k is greater than current average fitness of the group, the subregion k will be refined.

2) Update regional attribute. If current average fitness of individuals in subregion k is smaller than current average fitness of the group, and at the same time, the fitness of the best individual in subregion k is smaller than current average fitness of group, the subregion k will not be refined. Then the regional attribute of all the subregions are updated according to (1) and (2).

2.3 Solution Updating

The belief space is used to conduct generating solutions at next generation. Each individual has its own individual belief space.

There are 4 situations in which the methods of generating solution at next generation is different:

1) If $\begin{cases} \text{obj}_i[gen] < obj_{ave}[gen] \\ \text{obj}_i[gen] < obj^{local}_i \end{cases}$, individual i will be conducted by its own individual belief space at next generation.

2) If $\begin{cases} \text{obj}_i[gen] < obj_{ave}[gen] \\ \text{obj}_i[gen] > obj^{local}_i \end{cases}$, and $cnt > mCnt$, the individual i will be conducted by population belief space at next generation; but if $cnt < mCnt$, it will still be conducted by its own individual belief.

3) If $\begin{cases} \text{obj}_i[gen] > obj_{ave}[gen] \\ \text{obj}_i[gen] < obj^{local}_i \end{cases}$, individual i at next generation will be conducted by its own individual belief space.

4) If $\begin{cases} \text{obj}_i[gen] > obj_{ave}[gen] \\ \text{obj}_i[gen] > obj^{local}_i \end{cases}$, the individual i at next generation will be conducted by the population belief space.

2.4 Update Pheromone

Concentration When ants group finish their search and all the solutions have been evaluated, the pheromone concentration of the paths will be updated as:

$$\tau_{i,i+1}[gen+1] = \rho\tau_{i,i+1}[gen] + \sum_{i\in ants} \Delta\tau_i \tag{3}$$

$$\Delta\tau_i = Q/ftn(sln^{gen}_i) \tag{4}$$

2.5 The steps of the algorithm

Step 1 Initialize pheromone concentration, and parameters. Ants group are set to S for iteration. gen is set to 1.

Step 2 Ants search regular solutions, and then evaluate these solutions. Update the global best solution $globalBest$.

Step 3 Update global pheromone concentrations.

Step 4 Update the two belief space.

Step 5 Generate new solutions. $gen = gen + 1$. Then evaluate these solutions, and update the global best solution $globalBest$.

Step 6 If $gen = maxGen$, output $globalBest$; or else return to **Step 3**.

3 Tests and Analysis

3.1 Common standard functions

The dimension n of each function [11] is given a value of 30, x^* is the theoretical global optimal solution.

(1) Sphere Model

$$\min f_1(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2$$

$S = [-100, 100]^n, \boldsymbol{x}^* = (0,0,...,0)^T, f_1(\boldsymbol{x}^*) = 0.$

(2) Cosine Mixture Problem

$$\min f_2(\boldsymbol{x}) = \sum_{i=1}^{n} x_i^2 - 0.1 \sum_{i=1}^{n} \cos(5\pi x_i)$$

$S = [-1, 1]^n, \boldsymbol{x}^* = (0,0,...,0)^T, f_2(\boldsymbol{x}^*) = -0.1n.$

(3) Exponential Problem

$$\min f_3(\boldsymbol{x}) = -(\exp(-0.5 \sum_{i=1}^{n} x_i^2))$$

$S = [-1, 1]^n, \boldsymbol{x}^* = (0,0,...,0)^T, f_3(\boldsymbol{x}^*) = -1.$

(4) Griewank Problem

$$\min f_4(\boldsymbol{x}) = 1 + \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos(\frac{x_i}{\sqrt{i}})$$

$S = [-600, 600]^n, \boldsymbol{x}^* = (0,0,...,0)^T, f_4(\boldsymbol{x}^*) = 0.$

(5) Levy and Montalvo Problem

$$\begin{aligned}\min f_5(\boldsymbol{x}) = {} & 0.1\{\sin^2(3\pi x_1) \\ & + \sum_{i=1}^{n-1} (x_{i-1})^2[1 + \sin^2(3\pi\, x_{i+1})] \\ & + (x_{n-1})^2[1 + \sin^2(2\pi x_n)]\}\end{aligned}$$

$S = [-5, 5]^n, \boldsymbol{x}^* = (1,1,...,1)^T, f_5(\boldsymbol{x}^*) = 0.$

(6) Schwefels Problem

$$\min f_6(\boldsymbol{x}) = \sum_{i=1}^{n} (\sum_{j=1}^{i} x_j)^2$$

$S = [-100, 100]^n, \boldsymbol{x}^* = (0,0,...,0)^T, f_6(\boldsymbol{x}^*) = 0.$

(7) De Jong Function

$$\min f_7(\boldsymbol{x}) = \sum_{i=1}^{n-1} 100(x_{i+1} - x_i^2)^2 + (x_{i+1} - 1)^2$$

$S = [-10, 10]^n, \boldsymbol{x}^* = (1,1,...,1)^T, f_7(\boldsymbol{x}^*) = 0.$

(8) Sinusoidal Problem

$$\min f_8(\boldsymbol{x}) = -[2.5 \prod_{i=1}^{n} \sin(x_i - \frac{\pi}{6}) + \prod_{i=1}^{n} \sin 5(x_i - \frac{\pi}{6})]$$

$S = [0, \pi]^n, \boldsymbol{x}^* = (\frac{2\pi}{3}, \frac{2\pi}{3}, ..., \frac{2\pi}{3})^T, f_8(\boldsymbol{x}^*) = -3.5.$

3.2 Results and analysis

The algorithm is compared with the prototype culture algorithm (CA), ant colony algorithm (abb. **ACO**), maximal-minimal ant colony algorithm (abb. **MMACO**), and elite ant colony algorithm (abb. **EACO**). Some similar parameters of these algorithms are set same value, for the comparability of numerical tests: *ants* is set to 20, which is the number of ants group, and the group size of CA is also set to 20; *maxGen* is set to 1000, which is the maximal times of iteration; Q is set to 1.0, which is the initial value of pheromone concentration on each path; the volatile coefficient of pheromone ρ is set to 0.2; maximal-minimal pheromone concentration is set to (0.2, 1.0). The binary bits of each variable are set to 20. All the algorithms run 50 times separately, for each test function. The times that the algorithm was convergent (abb. **TOC**) and the average generation at which the algorithm first time found the global best solutions (abb. **AFG**) are shown in Table 3.1.

Table 3.1 Numerical tests results

	CACO		CA		ACO	
	TOC	AFG	TOC	AFG	TOC	AFG
$f_1(\boldsymbol{x})$	50	189	16	688	15	530
$f_2(\boldsymbol{x})$	50	215	11	712	12	616
$f_3(\boldsymbol{x})$	50	206	14	825	12	598
$f_4(\boldsymbol{x})$	50	221	10	738	11	863
$f_5(\boldsymbol{x})$	50	228	10	891	12	901
$f_6(\boldsymbol{x})$	50	253	9	942	10	922
$f_7(\boldsymbol{x})$	50	268	9	739	8	762
$f_8(\boldsymbol{x})$	50	226	12	822	12	721
	MMACO		EACO			
	TOC	AFG	TOC	AFG		
$f_1(\boldsymbol{x})$	22	335	23	378		
$f_2(\boldsymbol{x})$	19	531	19	421		
$f_3(\boldsymbol{x})$	21	457	22	536		
$f_4(\boldsymbol{x})$	19	556	21	457		
$f_5(\boldsymbol{x})$	20	582	18	764		
$f_6(\boldsymbol{x})$	14	537	16	592		
$f_7(\boldsymbol{x})$	14	628	13	575		
$f_8(\boldsymbol{x})$	18	533	21	468		

Remark: **AFG** did not count times that algorithm was not convergent.

As shown in Table 3.1, CA and ACO are not good at high dimensions and complex optimization problems. Their convergence rates are about 25%. It is indicated that only small probability to find the global best solutions when solution space is huge and the dimensions are high through random search. MMACO limits the available scope of pher-omone concentration, and has obtained a certain effect on restraining premature and improving global search, but the convergence rate is still unsatisfactory, 44% approximately. It is indicated that only improving the representation of pheromone concentration may not improve performance of algorithms satisfyingly. EACO relies on elite ants to provide the guidance at whole group, and promote the convergence rate to 45%. But when elite ants are not excellent enough, the effectiveness of the algorithm is discounted. CACO adopts the method that using the individual belief space to guide better individuals for deep search, and the population belief space to guide worse ones for broad search, so it acquires the best performance on convergence rate and convergence speed.

3.2.1 Influence analysis of the parameter

The influence of *mCnt* on the performance of CACO is analyzed in this part. *mCnt* was set to different values, and all the results are shown in Table 3.2.

Table 3.2 Convergence speed at different *mCnt*

		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$
mCnt	10	322	465	420	406	418
	20	238	401	388	325	351
	30	189	215	206	221	228
	40	286	352	385	324	372
		$f_6(x)$	$f_7(x)$	$f_8(x)$		
mCnt	10	423	436	408		
	20	324	372	365		
	30	253	268	226		
	40	298	325	373		

When *mCnt* is set too small (as an example: 10), the times that individuals are guided by population belief are increased, yet the depth of individuals search is limited. And so the convergence speed of CACO is reduced. When *mCnt* is set too big (as an exaple: 40), the impediment is also obvious. In this situation, the probability for worse individuals drop-ing their bad solution space and jumping to a better one is low. 30 may be the most suitable value for *mCnt* in CACO.

3.2.2 Influence analysis of the population belief

In this part, the influence of the population belief space is analyzed. All the individuals are guided by their own individual belief space only, and with this strategy a new algorithm NGA is found. The results of NGA on test functions above are shown in Table 3.3. The parameters are set the same as CACO.

Table 3.3 The performance of NGA

		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$
NGA	TOC	26	12	21	16	16
	AFG	352	681	831	764	623
		$f_6(x)$	$f_7(x)$	$f_8(x)$		
NGA	TOC	17	13	14		
	AFG	547	611	705		

It is indicated that without the population belief space guidance, the convergence rate of NGA is reduced. In fact, NGA degenerates to quasi-Newton method, and as known, the latter is not good at high dimensions optimization.

3.2.3 Influence analysis of the individual belief

In this part, the influence of the individual belief space is analyzed. All the individuals are guided by population belief space only, and with this strategy another new algorithm NIA is found. The results of NIA on test functions above are shown in Table 3.4. The parameters are set the same asCACO.

Table 3.4 The performance of NIA

		$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$
NIA	TOC	29	18	22	16	20
	AFG	469	806	887	863	732
		$f_6(x)$	$f_7(x)$	$f_8(x)$		
NIA	TOC	18	15	18		
	AFG	771	783	811		

It is indicated that without the individual belief guidance, the convergent rates of NIA is reduced either, but yet higher than NGA. For the population belief space guidance, the variety of individuals in NIA is guaranteed. These results present the effect of the population belief space in iteration process. The convergence speed of NIA is slower than NGA, and so obviously, the individual belief space has an important effect on the speed of optimization.

4 Conclusions

A new culture ant colony algorithm (CACO) is introduced in the paper for continuous optimization. The individuals in CACO, according to the rules, are guided by the individual belief space for deep search, and the population belief space for broad search. This mechanism ensures that CACO has excellent ability on finding global best solutions and fast optimization. The testing results on eight common standard functions, compared with four other algorithms prove the views above. Finally, the influence and effect of the parameter, the individual belief space and the population belief space are analyzed. It is indicated that CACO is the best one in these algorithms.

Acknowledgement

This research was supported by National Natural Science Foundation of China 'Intersections Group Coordination Control Theories and Methods Research based on Group Dynamics' (No. 61174184).

References

[1] A. Hertz and S. Kobler, A framework for the description of evolutionary algorithms, European Journal of Operational Research **126**, 1-12 (2000).

[2] W. Ma and Q. Zhu, Fast Continuous Ant Colony Optimization Algorithm for Solving Function Optimization Problems, Acta Electronica Sinica **36**, 2120-2124 (2008).

[3] J. Gao and R. Shan, A New Method for Modification Consistency of the Judgment Matrix Based on Genetic Ant Algorithm, Applied Mathematics Information Sciences **6**, 35-39 (2012).

[4] H. Wang, H. Xu, S. Yi and Z. Shi, A tree-growth based ant colony algorithm for QoS multicast routing problem, Expert Systems with Applications **38**, 11787-11795 (2011).

[5] I. Khodashinskii and P. Dudin, Identification of fuzzy systems using a continuous ant colony algorithm, Optoelectronics, Instrumentation and Data Processing **48**, 54-61 (2012).

[6] H. Lee and H. Tseng, A hybrid system for facility layout by integrating simulation and ant colony optimization, Applied Mathematics Information Sciences **6**, 387-396 (2012).

[7] H. Zhang, J. Zhou, Y. Zhang, Y. Lu and Y. Wang, Culture belief based multi-objective hybrid differential evolutionary algorithm in short term hydrothermal scheduling, Energy Conversion and Management **65**, 173-184 (2013).

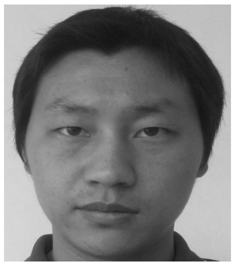
[8] C. Soza, R. Becerra, M. Riff and C. Coello, Solving timetabling problems using a cultural algorithm, Applied Soft Computing **11**, 337-344 (2011).

[9] Y. Hsieh, P. You, A New Space-Filling Curve Based Method for the Traveling Salesman Problems, Applied Mathematics Information Sciences **6**, 371-377 (2012).

[10] Y. Guo, D. Liu, J. Cheng and H. Wang, Adaptive cultural algorithm adopting mixed mutation, Tien Tzu Hsueh Pao / Acta Electronica Sinica **39**, 1913-1918 (2011).

[11] D. Kusum and T. Manoj, A new mutation operator for real coded genetic algorithms, Applied Mathematics and Computation **193**, 211-230 (2007).

Jianmin Xu is a Professor at School of Civil Engineering and Transportation, South China University of Technology. He got the doctoral degree at 1994. His research interest covers control theory and control engineering, intelligent optimization and application, and intelligent transportation systems.



Minjie Zhang received the Master Degree in Engineering from Guangdong University of Technology in 2010, and now is studying for a doctor's degree of Traffic Information Engineering and Control at South China University of Technology. His research interests are intelligent algorithm, network optimization and traffic control systems.

Yanguang Cai is a Professor at Faculty of Automation, Guangdong University of Technology. He got the doctoral degree at 1996. His research interest covers complex network system modeling, logistic control and optimization, intelligent transportation system, intelligent optimization, the thing network information processing and optimization control.