International Journal of Information Technology, Vol.12, No.3, 2006

A Novel Ant Colony System Based on Delauney Triangulation and Self-adaptive Mutation for TSP

Chao-Xue Wang, Du-Wu Cui, Yi-Kun Zhang, Zhu-Rong Wang

School of Computer Science and Engineering,
Xi'an University of Technology, Xi'an, 710048 China
Wbllw@126.com

Abstract

A novel ant colony system which employs a candidate set strategy based on Delaunay triangulation (CSDT) and a self-adaptive mutation operator (SMO) for TSP (DSMACS) is proposed. Under the condition that all the edges in the global optimal tour are nearly all contained in the candidate sets, CSDT can limit the selection scope of ants at each step to average six cities below, and thus substantially reduce the size of search space. To the shortage that search is possibly trapped in local optimal regions owing to the locality of this candidate set, SMO, which can self-adaptively adjust the size of neighborhood search scope of mutation operator, is designed to improve the global search ability of DSMACS by combining inversion and inserting mutation operator in genetic algorithm. The Simulation of TSP shows DSMACS can not only greatly increase the convergence speed but also avoid the premature convergence phenomenon effectively.

Keywords: ant colony system; candidate set strategy; Delauney Triangulation; self-adaptive mutation; TSP

1 Introduction

The Traveling Salesman Problem (TSP), where the task is to find the shortest closed tour through a given set of *n* cities with known inter-city distances such that each city is visited exactly once and the tour ends at the start city, is a well-known NP-hard problem [1]. Not only is TSP broadly applicable to a variety of routing and scheduling problem, but it is also usually considered as a standard test-bed for novel algorithmic ideas such as simulated annealing, tabu search, evolutionary algorithms including genetic algorithm, ant colony optimization (ACO) and so on, among which ACO inspired by the foraging behavior of real ant was first introduced by Dorigo and his colleagues [2, 3, 4] and has become one of the most efficient algorithms for TSP [3, 5].

It is a perpetual topic in the research field of meta-heuristic how to improve the convergence speed under the condition of guaranteeing the solution quality, and ACO is no exception. Since ACO is a constructive meta-heuristic and at each step ants consider the entire set of possible elements before choosing just one, the vast majority of an ant algorithm's runtime is devoted to evaluating the utility of reachable elements. So, both comparative slow convergence speed and comparative long runtime are the quite prominent problems in ACO. In order to solve these problems, the candidate set strategy that ants select from the narrowed set first and only if there are no feasible candidates are the remaining cities considered, which can limit this selection scope of ants to a narrowed set, is rifely adopted in ACO. The most commonly used candidate set strategy for TSP is the candidate set strategy based on nearest neighbor (CSNN), in which a set of the k nearest cities is maintained for each city. For example, some algorithms use CSNN (k=15 or k=20) [3, 5], some use CSNN (k= n/w, n is the total number of cities for TSP, w is a variable parameter varying with n) [6]. When candidate set strategy is used in ACO, the total number of edges in the global optimal solution contained in candidate sets and the limiting extent of candidate set to the selection scope of ants, which vary with the difference of candidate set strategy, will influence the performance of ACO.

By applying a candidate set based on Delauney triangulation (CSDT) and a self-adaptive mutation operator (SMO) to ant colony system (ACS) [3], a novel ant colony system (DSMACS) is presented. The theoretical analysis and experimental results for 16 instances of TSP demonstrate that CSDT can limit the search scope of ants at each step to average six cities below under the condition that all edges in the global optimal tour are nearly all contained in the candidate sets and is superior to CSNN (k=20). The mechanism and algorithm of SMO, which can self-adaptively adjust the size of neighborhood search scope of mutation operator and prevent DSMACS from being trapped in local optimal regions, are given. The simulation of TSP shows that MMACS can not only increase convergence speed greatly but also avoid the premature convergence phenomenon efficiently.

2 The Ant Colony System

The first ACO algorithm proposed is ant system (AS)[2], and since then Dorigo and other researchers have introduced many improved ACO algorithms based on AS, among which ant colony system (ACS)[3] has better performance and is a representative of ACO. The sketch of ACS can be shown in Fig.1.

```
Initialize
Loop /* at this level each loop is called an iteration */
    Each ant is positioned on a starting node
    Loop /* at this level each loop is called a step */
        Each ant applies a state transition rule to incrementally build a solution and
                    a local pheromone updating rule
    Until all ants have built a complete solution
    A global pheromone updating rule is applied
Until End_condition
```

Fig. 1. The sketch of ACS

In the following, we take TSP as an example to explain ACS. For the purpose of convenient expression, we first give some symbols: *n* and *m* are the total number of cities and ants respectively, τ_0 is the initial pheromone level.

Initially, *m* ants are placed on *m* cities randomly chosen, and there is the same pheromone level τ_0 on each edge. Suppose that an ant *k* is on city *i* and will chooses the next city *j* to move to, well then, the city *j* can be confirmed by applying the following state transition rule:

$$j = \begin{cases} \arg\max_{u \in J_k(t)} \{ \tau_{iu}(t)[\eta_{iu}]^{\beta} \}, \text{ if } q \le q_0 \text{ (exploitation)} \\ J, \qquad \text{otherwise (biased exploration)} \end{cases} \quad (\mathbf{1})$$

where $\tau_{iu}(t)$ is the pheromone level on edge *(i, u)* at the *t*-th step, $\eta_{iu} = 1/d_{iu}$ is the inverse of the distance d_{iu} from city *i* to city *u*, $J_k(t)$ is the set of cities that remain to be visited by ant *k* at the *t*-th step, β is a parameter that determines the relative importance of pheromone versus distance($\beta > 0$), *q* is a random number uniformly distributed in [0…1], q_0 is a parameter ($0 \le q_0 \le 1$), and *J* is a random variable selected according to the probability distribution given in Eq.(2).

$$p_{ij}^{\ k}(t) = \begin{cases} \dfrac{\tau_{ij}(\mathrm{t}) \cdot [\eta_{ij}]^{\beta}}{\sum\limits_{u \in J_k(t)} \tau_{iu}(\mathrm{t}) \cdot [\eta_{iu}]^{\beta}}, if \ j \in J_k(t) \\ 0, \qquad \text{otherwise} \end{cases} \quad (\mathbf{2})$$

where $p_{ij}^{k}(t)$ is the probability with which ant *k* in city *i* chooses to move to the city *j* at the *t*-th step.

The state transition rule resulting from Eq. (1) and Eq. (2) is called pseudo-random-proportional rule. This state transition rule favors transitions toward cities connected by short edges and with a large amount of pheromone. The parameter q_0 determines the relative importance of exploitation versus exploration: every time an ant in city *i* has to choose a city *j* to move to, it samples a random number *q*. If $q \le q_0$ then the best edge, according to Eq. (1), is chosen (exploitation), otherwise an edge is chosen according to Eq. (2) (biased exploration).

While building a tour of the TSP, ants visit edges and change their pheromone levels by applying the local pheromone updating rule of Eq. (3).

$$\tau_{ij}(t+1) \leftarrow \tau_{ij}(t)(1-\rho) + \rho \Delta \tau_{ij} \quad (\mathbf{3})$$

where ρ denotes the local pheromone decay parameter($0 < \rho < 1$), $\Delta \tau_{ij} = \tau_0$ denotes the amount of pheromone deposited on edge *(i, j)*.

The role of the ACS local pheromone updating rule is to shuffle the tours, so that the early cities in one ant's tour may be explored later in other ants' tours. In other words, the effect of local pheromone updating is to make the desirability of edges change dynamically: every time an ant uses an edge this becomes slightly less desirable (since it loses some of its pheromone). In this way ants will make a better use of pheromone information: without local pheromone updating all ants would search in a narrow neighborhood of the best previous tour.

The global pheromone updating is performed after all ants have completed their tours. The pheromone level is updated by applying the global pheromone updating rule of Eq. (4).

$$\tau_{ij}(t+1) \leftarrow (1-\alpha)\tau_{ij}(t) + \alpha\nabla\tau_{ij} \quad \textbf{(4)}$$

where

$$\nabla\tau_{ij} = \begin{cases} (L_{gb})^{-1}, & \text{if edge}(i,j)\in\text{the best tour} \\ 0, & \text{otherwise} \end{cases}$$

$0<\alpha<1$ is the global pheromone decay parameter, and L_{ab} is the length of the best tour from the beginning of the trial.

The global pheromone updating is intended to provide a greater amount of pheromone to shorter tours. Eq. (4) dictates that only those edges belonging to the best tour will receive reinforcement.

3 A Novel Ant Colony System Based on Delaunay Triangulation and Self-adaptive Mutation for TSP

The sketch of DSMACS is shown in Fig.2, where Con_Max denotes the iterations when SMO can be applied. In the following, CSDT and SMO will be described in detail.

```
Initialize
Constructing CSDT
Loop /* at this level each loop is called an iteration */
    Each ant is positioned on a starting node
    Loop /* at this level each loop is called a step */
        Each ant applies a state transition rule with CSDT to incrementally build a
        solution and a local pheromone updating rule
    Until all ants have built a complete solution
    A global pheromone updating rule is applied
    SMO is applied to the current best solution when iterations >=Con_Max
Until End_condition
```

Fig. 2. The sketch of DSMACS

3.1 The Candidate Set Based on Delaunay Triangulation

Firstly, we give some basis knowledge about Delaunay triangulation as follows.

Definition 1 Triangulation**:** Suppose $p = \{p_i, i = 1,2,...,n\}$ is the set of points in a plane, let $\xi = \{\overline{p_i p_j} \mid p_i, p_j \in p, i \neq j\}$ be the set of line segments, then a graph $T(p) = (p_T, \xi_T)$ can be defined. Where $p_T = p$, ξ_T is the biggest subset of ξ and meanwhile satisfies the condition that there isn't any cross between its any two line segments, then *T* is defined as triangulation on *P*.

Definition 2 Delaunay triangulation**:** If a triangulation has the property that for its each edge we can find a circle containing the edge's endpoints but not containing any other points, this triangulation is defined as Delaunay triangulation.

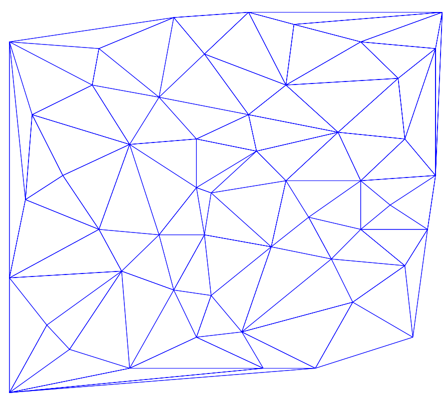


Fig. 3. The Delaunay triangulation of eil51.tsp in the plane

*Theorem 1***:** For a Delaunay triangulation made up of *n* points, if there are n_0 points in its convex hull, then it has $3n- n_0-3$ edges and $2n- n_0-2$ triangles.

Proof: Suppose *t* is the total number of triangles, *e* is the total number of edges, then except for the edges on the convex null, all the edges inside the convex null are shared two triangles, so $2e- n_0=3t$ holds. Meanwhile, according to the Ruler Formula, $n-e+ (t+1) =2$ holds. From the two equations above, we have $e = 3n- n_0 – 3$ and $t= 2n- n_0 – 2$.

Definition 3 the candidate set based Delaunay triangulation: all cities of the other endpoints of all edges incident to each city in the Delaunay triangulation of TSP make up of the candidate set of this city.

A comparison of the percentage of their edges shared with the global optimal tour for 16 instances of TSP from TSPLIB between CSDT and CSNN (*k*=20) is shown in Fig.4, where their average percentage is 99.58% and 99.52% respectively. Hereby the two kinds of candidate set are approximately equal in quality, but their limiting extent to the selection scope of ants at each step is obviously difference. According to the Theorem 1, if the total number of cities in TSP is *n*, then all the edges in Delaunay triangulation is $3n$- n_0-3(n_0 is the total number of cites in convex null) and the size of candidate set of each city is below six. So the limiting extent of CSDT to the selection scope of ants at each step is bigger than CSNN (*k*=20). In addition, every two edges in Delaunay triangulation impossibly cross, and this is corresponding with the property that there is no cross in the global optimal tour. But CSNN (*k*=20) hasn't this merit.

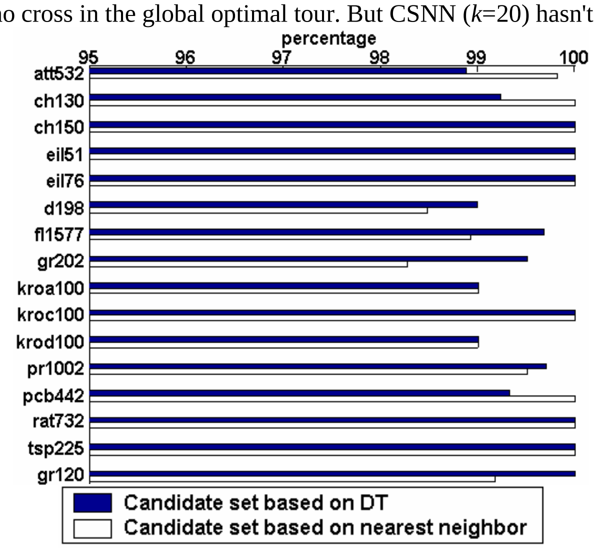


Fig. 4. A comparison of the quality of two kinds of candidate set

Of course, constructing CSDT creates an overhead, but this overhead can be accepted since its time complexity can be limited to $O(N \log N)$ according to the algorithm of Ref. [7]. Generally speaking, CSDT is superior to CSNN (*k*=20).

3.2 A Self-adaptive Mutation Operator

If CSDT is adopted in DSMACS, this algorithm will quickly find out local optimal solutions. But it is likely to take comparative long time to find out the global optimal solution, and sometimes this algorithm can only converge to a local optimal solution.

As we all know, mutation operator can keep the diversity of population to prevent search from being trapped in local optimal regions in genetic algorithm. The commonly used mutation operators for TSP have inversion and insertion operator [8], whose neighborhood search scopes are two edges and tree edges respectively. A measure of search difficulty, fitness distance correlation (FDC) is introduced in Ref. [9], and there exists a high and positive FDC for TSP, which indicates that the smaller the solution cost is, the closer are the solutions – on average – to the global optimal solution [10]. Hereby there is often a small quantity of different edges between a local optimal solution and the global optimal solution or a better local optimal solution. In other words, we possibly find out a better local optimal solution or the global optimal solution so long as a small quantity of edges is changed. On the other hand, a local optimal solution with respect to one mutation operator is not necessary so for another owing to the different neighborhood search scopes, but the global optimal solution with respect to all mutation operators are the same one.

Based on the consideration above, SMO is designed by combining inversion and inserting mutation operator in genetic algorithm in this paper. This operator can self-adaptively adjust the size of neighborhood search scope of mutation operator, keep the diversity of solutions and improve the global search ability of DSMACS efficiently. In DSMACS, when the algorithm has evolved for many iterations and the search will possibly be in stagnation, this operator is applied to the current best solution. Suppose the search will possibly be in stagnation and the current best tour is ($b_1b_2\ldots b_i\ldots b_j\ldots b_n$) after the algorithm has evolved for Con_Max iterations, then the pseudo-code of this operator's algorithm can be given in the following.

```
Begin
  for(i=1;i<n-1;i++)
  {
    for(j=i+2;j<n+1;j++)
    {
     if (j-i=n-1)break;
     if (after the path between b_i and b_j is reversed,
         tour's length is reduced)
     {
      reverse the path between b_i+1 and b_j;
      continue;
     }
     else if (after city b_j is inserted behind city b_i,
              tour's length is reduced)
     {
      insert city b_j behind city b_i ;
     }
    }
  }
  End
```

4. The Simulation and Analysis

DSMACS is realized in VC6.0 and run on a PC (CPU Pentium4 2.4GHz, 256 MB memory) with Windows 2000 Operating System. Many instances of TSP from TSPLIB are used in simulation, and perfect results are achieved. In all experiments the numeric parameters, except when indicated differently, are set to the following values: $m=3\sim10$, $Con\_Max=3\sim15$, $\beta = 2$, $q=0.9$, $a = 0.1 \sim 0.2$, $r = 0.1 \sim 0.2$, $\tau_0 =(nL_{nn})^{-1}$.(L_{nn} is the tour length produced by the nearest neighbor heuristic [11]).

4.1 The Comparison between DSMACS and the Algorithm in Reference

An ant colony optimization algorithm based on mutation and dynamic pheromone updating (NDMACO), which adopted CSNN ($k = n/w$, n is the total number of cities for TSP, w is a variable parameter varying with n) and a unique mutation scheme, was introduced in Ref. [6]. In order to compare DSMACS with NDMACO, some instances of TSP that are the same as ones used in NDMACO are chose in our simulation. A comparison of the final solution and convergence number and a comparison of convergence time between DSMACS and NDMACO are shown in Table.1 (the results of NDMACO are directly taken from Ref. [6]) and Table.2 respectively. It can be seen that DSMACS not only finds out the global optimal solutions for the four instances of TSP but also has very quick convergence speed while NDMACO only finds out the local optimal solutions for Pr107 and D198.

Table 1. A comparison of the final solution and convergence number between DSMACS and NDMACO

Name	Best length of DSMACS	Best length of NDMACO	Convergence number of DSMACS	Convergence number of NDMACO
Eil51	426	426	5	7
Berlin52	7542	7542	4	11
Pr107	44303	44383	33	330
Dl98	15780	15796	243	800

Table 2. A comparison of convergence time between DSMACS and NDMACO

Name	Convergence time of DSMACS(s)	Convergence time of NDMACO(s)
Eil51	0.01	0.02
Berlin52	0.01	0.02
Pr107	0.08	0.6

4.2 The Comparison between DSMACS and ACS

DSMACS and ACS are tested on five hard and large-scaled TSP respectively, and the experimental results are shown in Table.3, where each experiment consists of at least 20 trials and the experimental results of ACS are directly taken from Ref. [3]. It can be seen from Table.3 that DSMACS makes not only the quality of solutions better than ACS but also the speed of convergence hundreds of times faster than ACS.

Table 3. A comparison between DSMACS and ACS

Name	Optimum (1)	DSMACS best integer length (2)	DSMACS number of tours generated to best	DSMACS average integer length	Relative error $\frac{(2)-(1)}{(1)}$	ACS best integer length (3)	ACS number of tours generated to best	ACS average integer length	Relative error $\frac{(3)-(1)}{(1)}$
D198	15780	15780	1458	15785	0%	15888	585000	16054	0.68%
Pcb442	50779	50979	2100	51180	0.39%	51268	595000	51690	0.96%

Att532	27686	27705	3500	27729	0.07%	28174	830658	28523	1.67%
Rat783	8806	8860	4300	8893	0.61%	9015	991276	9066	2.37%
Fl1577	22249	22453	15890	22560	0.92%	22977	942000	23163	3.48%

4.3 The Analysis of Diversity

When DSMACS is used for solving d198.tsp, the variation of tour length with iteration is shown in Fig.5. It can be seen that the best solution has verged on local optimal solutions after 18 iterations and then the global optimal solution is obtained after SMO is adopted and DSMACS have evolved about 220 iterations. In the meantime, the average tour length has been in shaking randomly all the time after 18 iterations and can have kept certain distance to the best solution. Therefore, DSMACS can greatly reduce the size of search space and improve the convergence speed after CSDT is adopted; meanwhile SMO can self-adaptively adjust the size of neighborhood search scope and maintains the diversity of solutions. This is one of the reasons why the DSMACS have the satisfying ability of global optimization while improving the convergence speed greatly.

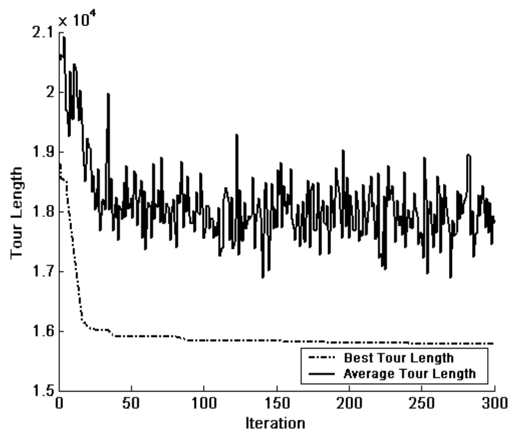


Fig. 5. The variation of best tour length and average tour length with iteration

5. Conclusions

On the basis of in-depth investigation into the domain knowledge of TSP, CSDT is designed. Although the global optimal solution is not a subset of Delauney triangulation [12], CSDT contains the edges of the global optimal solution at high probability. CSDT can make the selection scope of ants at each step reduce from *n (n-1)/2* to *6n* below, thus the runtime of DSMACS is reduced greatly and the convergence speed is improved greatly. In the meantime, in order to prevent

DSMACS from being trapped in local optimal regions caused by the locality of CSDT, SMO, which can self-adaptively adjust the size of neighborhood search scope of mutation operator, is designed by combining inversion and inserting mutation operator in genetic algorithm to improve the global search ability of DSMACS. The simulation tests of TSP show that DSMACS can restrain premature convergence phenomenon effectively during the evolutionary process while greatly increasing the convergence speed.

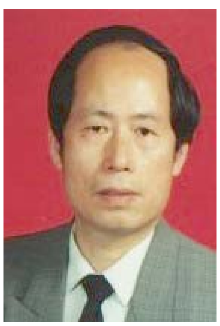
One of the main ideas underlying DSMACS is hybridization of meta-heuristics [13], by which mutation operator in genetic algorithm is introduced to DSMACS. Another idea is to give full play to the role of guidance function of domain knowledge [14], namely, the Delauney triangulation and the fitness landscape of TSP [9, 10, 12], by which CSDT is obtained and SHO is designed. Although DSMACS takes TSP as an example for explaining its mechanism, its idea can be used for other related algorithms.

References

1. Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G., Shmoys, D.B.: The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization. John Wiley and Sons, New York (1985)
2. Colormi, A., Dorigo, M., Manieaao, V.: Distributed Optimization by Ant Colonies. In Varela, F. and Bourgine, P. (eds.): Proc of the First European Conf On Artificial Life. Paris, France, Elsevier Publishing (1991) 134–142
3. Dorigo, M., Gambardella, L. M.: Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem. IEEE Trans on Evolutionary Computation (1997) 1(1) 53–66
4. Dorigo, M. and Di Caro, G.: The Ant Colony Optimization Meta-heuristic. In Corne, D., Dorigo, M., Glover, F. (eds.): New Ideas in Optimization. McGraw-Hill, London, UK (1999) 11–32
5. Stützle, T., Hoos, H.: MAX-MIN Ant System and Local Search for the Traveling Salesman Problem. In: IEEE Int'l Conf. on Evolutionary Computation. Indianapolis: IEEE Press (1997) 309–314
6. Zhu, Q. B., Yang, Z. J.: An Ant Colony Optimization Algorithm Based on Mutation and Dynamic Pheromone Updating. Journal of Software (2004) 15(2) 185–192 (In Chinese)
7. Wu, X. B., Wang, S. X., Xiao, C. S.: A New Study of Delaunay Triangulation Creation. ACTA GEODAETICA et CARTOGRAPHICA SINICA (1999) 28(1) 28-35(In Chinese)
8. Baeck, T., Fogel, D., Michalewicz, Z.: Evolutionary Computation 1: Basic Algorithms and Operators. Institute of Physics Publishing, Bristol, UK (2000)
9. Jones, T. and Forrest, S.: Fitness Distance Correlation as A Measure of Problem Difficulty for Genetic Algorithms. In Eshelman, L.J. (eds.): Proceedings of the 6th International Conference on Genetic Algorithms. Morgan Kaufman, San Francisco, CA, USA (1995) 184–192
10. Merz, P. and Freisleben, B.: Fitness Landscapes and Memetic Algorithm Design. In Corne, D., Dorigo, M., Glover, F. (eds.): New Ideas in Optimization. McGraw-Hill, London, UK (1999) 245-260
11. Rosenkrantz, D. J., Stearns, R. E. and Lewis, P. M.: An Analysis of Several Heuristics for the Traveling Salesman Problem. SIAM Journal on Computing (1977) 6(3) 563–581

12. Kantabutra, V.: Traveling Salesman Cycles are not Always Subgraphs of Voronoi Duals. Information Processing Letters (1983) 16(1) 11–12
13. Blum, C. and Roli, A.: Metaheuristics in combinatorial optimization: Overview and Conceptual Comparison. ACM Computing Surveys (2003) 35(3) 268–308
14. Wolpert, D. H., Macready, W. G.: No free lunch theorems for optimization. IEEE Trans on Evolutionary Computation (1997) 1 (1) 67– 82

Chao-Xue Wang received the B.S degree from Shaanxi Normal University, Xi'an, China in 1991 and the M.Eng's degree in Hohai University, Nanjing, China in 2003. After an appointment as an assistant professor of physics at Xi'an University of Architecture and Technology in 1992, he moved to Jiangsu Province and became a Senior Programmer and Senior Engineer in information science. He is currently working toward the Ph.D. degree in Xi'an University of Technology, Xi'an, China. His research interests include Intelligent Computing, Power Electronics and Electric Drive, and Electronic Commerce, etc.



Du-Wu Cui received his B.Eng's degree and M.eng's degree from Xi'an Jiaotong University, Xi'an, China in 1967 and 1983 respectively. He has been professor and Ph.D. supervisor in Xi' an University of technology, Xi'an, China. His research interests include AI, Multimedia Technology and Decision Support System (DSS), etc.