International Journal of Hybrid Information Technology
Vol.7, No.2 (2014), pp.227-236
http://dx.doi.org/10.14257/ijhit.2014.7.2.21

A Clustering Algorithm Based on Multi-agent Meta-heuristic Architecture

Zhu Qiang[1], Shun Yuqiang[2] and Cen Yang[1]

[1]*Zhejiang University of Media and Communications, Hangzhou 310018, P.R. China*
[2]*Changzhou University, Changzhou 213022, P.R. China*

zq.hz@163.com

Abstract

A clustering algorithm is proposed in this paper, which is based on discussion of multi-agent meta-heuristic architecture of the ant colony optimization algorithm. The multi-agent architecture of ant colony optimization meta-heuristic includes three levels. Level-0 agents build solutions, level-l agents improve solutions and level-2 agents update pheromone matrix. The updated pheromone then provides feedback information for the next iteration of solution construction. Mutation probability p and pheromone resistance ρ are the adaptive parameters, which can be adjusted automatically during the evolution progress. With the adaptive variable, the algorithm can solve the contradiction between convergence speed and precocity and stagnation. The algorithm has been tested and compared with the clustering algorithm based on Genetic and Simulate annealing. Experimental results show that the proposed algorithm is more effective, and the clustering quality and efficiency are promising.

Keywords: *Clustering, Ant colony optimization, Multi-Agent, Meta-heuristic*

1. Introduction

Clustering is a process of grouping a set of physical or abstract objects into classes of similar objects, so that the data objects within the same cluster are similar to each other and are dissimilar to the objects in other clusters.

Clustering plays an important role in data mining applications such as scientific data exploration, information retrieval, spatial database applications, web mining, client relationship management, medical diagnostics, computational biology, and many others.

There exists a multitude of clustering methods, which vary from the principles of the algorithm to their basic properties, such as the data manipulated, the shape of the clusters, the form of the final partitioning or the parameters that have to be provided, etc. Recently, some clustering algorithms based on meta-heuristics, such as tabu search [1], genetic algorithms [2], simulated annealing [3] and ant colony algorithm [4], have been proposed and successfully applied in some fields. The meta-heuristic techniques have been used to deal with combination explosion and NP-hard problems, which get favorable results. The properties of meta-heuristic involve global optimization, parallel, robustness, high efficiency and convenience to implement without special knowledge of domain.

Ant colony algorithm is a new bionic algorithm inspired from the behaviors of real ants in nature. A large number of ant based clustering models have emerged, since the movement of ants in real world is more close to the clustering process. Stigmergy is a concept to describe an indirect interaction between ants. It is known as a mechanism in which an individual reacts

ISSN: 1738-9968 IJHIT

to modification of its environment due to the action of another. The current ant based clustering model may be divided into two major categories according to different stigmergy.

The first category is inspired from ants organization of cemeteries, which is based on the Basic Model (BM) [4] proposed by Deneubourg *et al.* Lumer and Faieta generalized Deneubourg BM to apply it to data clustering analysis, which is called LF algorithm for short [5]. In this kind of model, it is the distribution of the objects in the environment that plays the role of stigmergic variable. The ants try to pick-up or drop objects on the two-dimensional-board through the perception of the objects spatial distribution, so that small clusters of items grow by attracting works to deposit more items. N. Monmarche et al. proposed a clustering algorithm based on ant cemetery named AntClass [6]. AntClass combines the stochastic principles of an ant colony with the deterministic principles of the K-means algorithm that can improve the convergence of the ant colony clustering. Wu *et al.* introduced another algorithm named CSI, which is derived from the basic model. The CSI has been successfully applied in customer behavior analysis and web document clustering [7, 8]. Another category of ant clustering model is based on ants foraging behavior. In this case, it is the pheromone deposited on the path when individual ants walk between a food source and their nest that plays the role of stigmergic variable. Dorigo *et al.* proposed the well-known Ant Colony Optimization (ACO) algorithm based on ants foraging behavior, and ACO has been used to solve several discrete optimization problems, such as TSP, Job shop, and so on [9, 10]. Some clustering models with ACO approach have been put forward recently, such as the new clustering method based on ant colony optimization reported by Yang *et al.* [11] and the ant colony approach for clustering described by Shelokar *et al.* [12].

In addition to the two categories of clustering models described above, Nicolas Labroche proposed an AntClust algorithm based on the chemical recognition system of ants [13]. H.Azzag reported an AntTree clustering algorithm based on the ants self-assembly behavior [14]. In summary, all ant colony-based clustering models are based on population-based meta-heuristic approach, with integration of individual random search and information positive feedback among individuals.

In this paper, a new clustering algorithm according to analysis on Multi-agent architecture of ant colony optimization algorithm is described in detail. The paper is organized as follows. Section 2 gives a discussion of ACO Multi-agent architecture. Section 3 describes the proposed algorithm in detail and section 4 reports the experimental results of evaluation and comparison of the algorithm's performance. The last section gives the conclusions and future work.

2. Multi-Agent Architecture of ACO Meta-heuristic

Ant colony optimization algorithms are multi-agent systems with population meta-heuristics. ACO algorithm can be abstractly seen as a multi-agent architecture containing three levels, which is showed in Figure 1 [15].

We define the agents "as the components consisting of direct or indirect communication function with specific structure and interface. As far as ACO concerned, agents are the ant individuals that hunt for problem solutions. The agents are organized in levels. In each level, one or more agents can act parallel. Agents belong to the same level have the same structure and same searching strategies. Agents belong to different levels can communicate to exchange results and information. The agents' functions of three levels are described as follows. (1) Level 0 (L0) agents are solution build agents. L0-agents provide initial solutions by using a constructive procedure biased by pheromone trails and heuristic information. The strategy of L0-agents employs actual constructive algorithms like random initialization, greedy construction, probabilistic (*e.g.*, pheromone based), *etc.* They provide a feasible solution or a set of solutions for the upper level. (2) Level 1 (Ll) agents' task is to improve the solution provided by L0. L1-agents implement some strategy (such as local search algorithm) and their recent past information (like short term memory) to intensify the search or to escape from local optima. The output of Ll is the improved solutions, which will be the input for Level 2. (3) Level 2 (L2) agents store the best solutions from L1 and then update the pheromone. The new updated pheromone is a feedback to be used again by LO-agents to construct new solutions. Consequently, iteration of the ACO algorithm can evolve the solution to proximate optimization gradually. Based on multi-agent architecture of ACO, we can construct new algorithms to settle different problems in various fields. In the rest of the paper, we will discuss a new clustering approach using multi-agent architecture of ACO in detail.

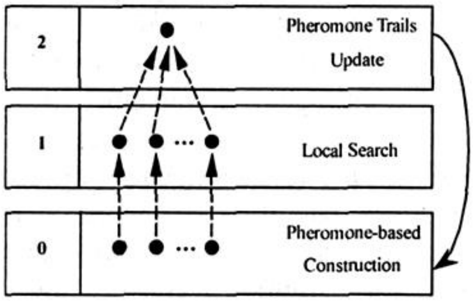

Figure 1. Multi-level architecture of ACO

3. Adaptive ACO Clustering Algorithm

3.1. Definition of clustering problem

Many clustering definitions have been proposed over years and the clustering problem is known to be a NP-complete problem [16]. In this work, the clustering problem is defined as an optimization problem. It is described as follows:

Instance: A finite set X , a distance measure $\delta(i,j) \in R_0^+$ for $i,j \in X$, two positive integers K and B, and a criterion function Je , (C, $\delta(.,.)$) on a K-partition $C = \{C, ..., C_K\}$ of X and measure $\delta(.,.)$.

Question: Is there a. partition of X into disjoint sets C_1 ,..., C_K such that J_e(C, $\delta(.,.)$).

Optimization: Find the partition of X into disjoint seta C_1 ,..., C_K that minimizes the expression Je (C, $\delta(.,.)$).

3.2. Coding and criterion function

Consider a given dataset X of N objects $\{x_1, x_2,..., x_N)$ in $\Re^n$ -dimensional space with R ants to build solutions. Let K be the cluster number, the clustering problem here is that R agents partition the dataset into K clusters to minimize the criterion function J_e (also called evaluation function in the following).

The first step is to code solution. Here a simple representative solution string is adopted. A solution of N objects is denoted by one string $s=\{c_1c_2\ldots c_n\}$, where $\{c_i, |i=1\ldots N\}$ is a cluster label of object i. If $c_i = c_j$, Object i and j are identical object. In sequel, the solution sets of R agents from a matrix $\mathrm{S}=\{s_1,s_2,\ldots,s_R\}$.

There are many criterion functions to evaluate the clustering result in clustering methods via cost optimization. The function adopted here is defined as the sum of squared Euclidean distances between each object and center of the belonging cluster, which is:

$$J_e = \sum_{j=1}^{K} \sum_{i=1}^{N} \mu_{ij}\, d(x_i, m_j) \qquad (1)$$

Where μ is an associated weight matrix of object x_i with cluster j, and $\mu_{ij} = \begin{cases} 1\ x_i \in j \\ 0 x_i \notin j \end{cases}$, m is a cluster center matrix of size $K \times N$ m_j is the average of the attribute values of samples in the cluster j, which can be computed as

$$m_j = \frac{\sum_{j=1}^{k} \sum_{i=1}^{N} \mu_{ij} x_i}{\sum_{j=1}^{K} \sum_{i=1}^{N} \mu_{ij}} \qquad (2)$$

The aim of clustering is to evaluate the solutions generated by ants repeatedly until the J_e value meets the satisfaction, which is J_e<*MinMeasure,* where *MinMeasure* is a predefined minimum function value.

Initialization needs to be implemented before the execution of the algorithm. The agents use the pheromone trail to assign a cluster label to each element of string *s*. Due to the problem of partition *N* objects into *K* clusters, a pheromone matrix *Ʈ* of size $N \times K$ is employed in this algorithm. The element value of pheromone matrix τ_{ij} denotes the pheromone concentration of object *i* associated to the cluster *j*. Thus, each object is associated with *K* pheromone concentrations. The trail values τ are initialized to the same values or zero before the execution of the algorithm, and the ants should start with empty solution strings.

3.3. Solution construction of Level 0

As preceding discussion on ACO multi-agent architecture, Level 0 is the basis of whole algorithm. The ant agents build solutions, which allocate each element of string s to a cluster label with the heuristic information combining random search. The process is: predefine the constructive probability $q(0<q<1)$, (a) according the pheromone matrix, select the cluster having the maximum pheromone concentration as object's cluster using probability q; (b) assign a cluster label using stochastic distribution with a mutation probability $p = 1-q$. The detail of L0 is written hereafter:

```
L0: for i=1 to R do
    Empty solution matrix S;
       Generate a set of random number
          $\{r_i \mid i = 1...N, 0< r_i< 1)$ corresponding to
          $s_i(c_1c_2...c_N)$ ;
       if $r_i > p$ then choose cluster with maximum
          pheromone concentration as $c_i$ ;
       else assign a cluster to $c_i$ arbitrarily within the defined $K$;
       Compute the evaluation function $F_i$ of $s_i$;
    end for
       Sort the values of evaluation function;
```

3.4. Local search of Level 1

In this level, the best solutions on the top L_{best}, can be improved further through local search. As above discussion, employing some form of local search can find better solution. Our strategy of local search is to alter the cluster number of each object in the solution string with certain threshold probability. Therefore, some new solutions are generated based on the original solution. The criteria function values of newly generated solution are computed using equation (1). If new solutions are better than old ones, the old solutions are replaced by the new solutions, otherwise, keep the old ones. Suppose the number of searching ant agents is L_{best}, $L_{best} \in [1,R]$, the detail of the L1 algorithm is described as follows:

```
LI: for k=l to $L_{best}$ do
        $s_h$=$s_k$;
        Change several cluster label of $s_h$ randomly;
        Compute the evaluation function value $F_h$ of $s_h$;
    if $F_h$<$F_k$ then $s_h \rightarrow s_k$, $F_h \rightarrow F_k$;
    end for
```

3.5. Pheromone updating with adaptive mechanism of Level 2

After performing the local search operation, the pheromone matrix is then updated depending on the quality of the solutions. Pheromone updating can fully make use of the dynamic information produced by the agents. Accordingly, the pheromone matrix is updated using the information produced by the previously found superior solutions at the end of the iteration. Then, the newly updated pheromone matrix is used again by L0-agents to construct new solutions in the next iteration. Suppose that there are L_{best} gents who generated best L_{best} solutions. These L_{best} agents imitate deposition of the pheromone trail of real ants by assigning τ_{ij} associated with the solution attributes, and evaporation as well. If $\rho \in [0,1]$ represents the persistence rate of the trail, then, (1-ρ) represents evaporation rate. The corresponding formula for pheromone updating is:

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{j=1}^{L_{best}} \Delta\tau_{ij}^{l} \qquad (3)$$

$$\Delta\tau_{ij}^{l} = \begin{cases} 1/F_l & if\ \ x_i \in j \\ 0 & otherwise \end{cases} \qquad (4)$$

where $\Delta\tau_{ij}^{l}$ is equal to $1/F_i$ and it is conversely proportional to the evaluation function value. That means the less evaluation function value the solution has, the more pheromone concentration information will be deposited for the i th element of the solution labeled by cluster j.

Generally, one of the most important issues in ACO meta-heuristic research is the balance between intensification (exploitation of the previous solutions) and diversification (exploration of the search space). These two factors are essential. As we know, intensification can intensify the superior solutions and accelerate convergence of the algorithm, while diversification can make the algorithm find better solutions. However, too much emphasis on the former can make agents converge to a local optimum and too much emphasis on the latter can cause an unstable state. In order to settle the contradiction between convergence speed and precocity and stagnation, the proposed algorithm introduces an adaptive mechanism to get the global optima with relatively faster convergence speed. The adaptive mechanism is to adjust the mutation probabilityρ and pheromone persistence rate ρ dynamically [17].

Mutation is a key factor that affects behavior of agents and algorithm performance. The higher the mutation probability is, the faster the new solutions can be generated. When value of ρ is too high, superior solutions also incline to be destroyed. On the contrary, if ρ is too low, it is hard to generate new solutions, leading to inefficient search process, and even stagnation. Similarly pheromone persistence rate is another important factor that is relevant to convergence speed and capability of global search. Higher value of ρ suggests that the agents speed up constructing new solutions, so as to intensify the capability of global search. At the same time slow down the convergence speed. If the value of ρ is too high, the condition is just on the contrary.

In order to solve the problem, the mechanism is adopted as follows. First, the algorithm stores the best solution after the ants completed their iterations once. If the best solution can't be improved after C times search cycles, the values of p and ρ are modified automatically. That is, $p(\mathrm{t})=p(t\text{-}1)+0.05$, $p_{\min}< p(t\text{-}1) < p_{\max}$, $p_{\min}$, $p_{\max}$, are predefined minimum and maximum values of mutation probability p , respectively; $\rho(\mathrm{t})= \rho(t\text{-}1)+0.01$, $\rho_{\min}< p(t\text{-}1) <\rho_{\max}$, $\rho_{\min}$, $\rho_{\max}$ are predefined minimum and maximum values of pheromone persistence rate, respectively. Therefore, the algorithm of L2 can be described as follows in detail.

```
L2: for i=l to N
        For j= 1 to K
            $\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \sum_{j=1}^{L_{best}} \Delta\tau_{ij}^{l}$
        end for j
    end for i
  if ( current best solution $\geq$ best solution before C times
            iterations ) then
    begin
    if $p_{\min}< p(t\text{-}1) < p_{\max}$ then $p(\mathrm{t})=p(t\text{-}1)+0.05$;
    if $\rho_{\min}< p(t\text{-}1) <\rho_{\max}$ then $\rho(\mathrm{t})= \rho(t\text{-}1)+0.01$;
    end if
        // The stopping condition of whole algorithm
        if ( $F_{\min}$ <MinMeasure or reach the max iteration number $M_c$ ) then output result;
            else Goto L0//Agents go on the next search iteration.
```

4. Experimental Results

We implement the proposed algorithm on lris dataset which is obtained from the UCI repository of machine learning. The dataset consists of 150 data object and each object has 4 attributes. The dataset can be classified into 3clusters. All algorithms are executed in C language. The parameters of algorithms are initialized as follows:R=50, p=0.1, $\tau_{ij}(0)$=0.01, $\rho(0)$=0.02, M_c=20000, The algorithms with adaptive mechanism and without adaptive mechanism are run for 10 times, respectively. The convergence curves are in Figure 2 and Figure 3.

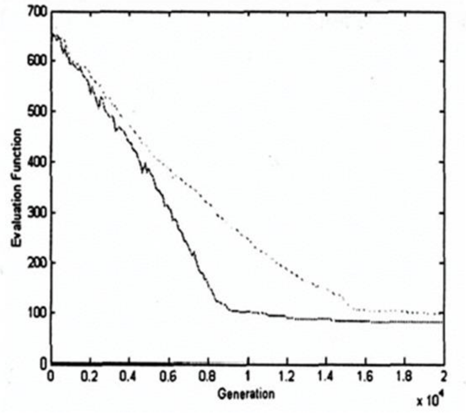

Figure 2. Convergence curves of non-adaptive ACO clustering

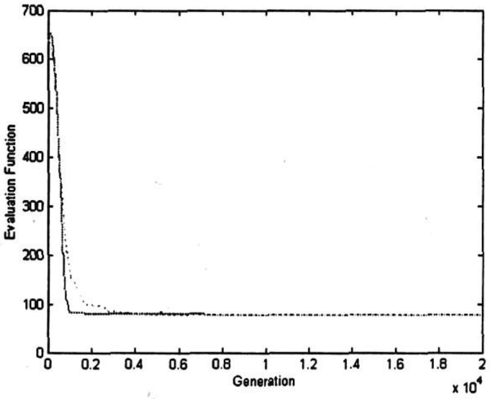

Figure 3. Convergence curves of adaptive ACO clustering

The solid and doc curves represent the best and the worst condition, correspondingly. It is showed that the adaptive ACO clustering algorithm can converge faster and reach better results, where the values of evaluation function are 78.940840 at each execution. The results are better than the non-adaptive ACO clustering algorithm whose average value of evaluation function are 84.2033521.

Table 1. Performance comparison on proposed algorithm, GA and SA

	Iris		Thyroid	
	Evaluation function	CPU Time(s)	Evaluation function	CPU Time(s)
Adaptive-ACO	92.257204	19	10001.13785	97
Non Adaptive-ACO	99.220427	53	101 12.62198	108
GA	132.73362	104	10137.92831	162
SA	97.100777	96	10114.04517	110

To evaluate the performance of the proposed algorithm, we also compare our algorithm with other meta-heuristic algorithms including genetic algorithm and simulated annealing algorithm on datasets Iris and Thyroid. Each algorithm runs 5 times respectively, and average results are given in Table 1. It is demonstrated that the results obtained by adaptive ACO clustering algorithm are superior to those of the GA and SA.

The time complexity of the proposed algorithm is also analyzed. The analysis is divided into three parts: L0, L1 and L2. L0 involves solutions construction and computation of formula (1) and (2), so its time complexity is $O(2RNK_n)$. Time complexity of Local search on L1 is $O(2\ L_{best},NKn)$. L2's time complexity is $O(NK)$, which execute pheromone updating. Generally speaking, the entire algorithm's complexity is proximately $O(M_{cnverge}RNK_n)$ in the worst condition, where $M_{cnverge}$ is the number of iteration needed for meeting the stopping condition. We can see that, the time complexity depends on the convergence situation to a great extent. With the problem size scaling up, the complexity increases linearly.

5. Conclusion

A new clustering algorithm based on multi-agent meta-heuristic architecture is proposed in this paper. The ACO meta-heuristic can be abstracted into a three-level framework. Each level comprises several agents. L0 is a solution construction level, L1 is a solution improvement level and L2 is a pheromone updating level with adaptive mechanism. The updated pheromone is used to construct new solutions in the next iteration as feedback information. Two parameters mutation probability and pheromone resistance, which can be adjusted automatically during the evolution progress, are involved in adaptive mechanism. With the adaptive variable, the algorithm can solve the contradiction between convergence speed and precocity and stagnation during the agents searching process. Experimental results show that the proposed algorithm is effective, and both of its clustering quality and efficiency is better- than the clustering algorithms based on Genetic and Simulate annealing. However, the proposed algorithm must define the number of clusters in advance. The efficiency and ability to solve larger size problems of clustering are still need to be improved as well. In summary, the meta-heuristic ACO for clustering is a challenging and active research field.

Acknowledgements

This work was supported by a grant from the National Natural Science Foundation of Zhejiang (No. Y1100314).

References

[1] C. S. Sung and H. W. Jin, “A tabu-search-based heuristic for clustering”, Patter Recognition, vol. 3, (**2010**), pp. 949-858.

[2] C. A. Murthy and N. Chowdhury, “In serch of optimal clusters using genetic algorithms Pattern Recognition Letters”, no. 17, (**2006**), pp. 825-832.

[3] D. E. Brown and C. L. Huntley, "A practical application of simulated annealing”, Pattern Recognition, vol. 25, no. 4, (**2002**), pp. 401- 412.

[4] J. L. Deneubourg, S. Goss, N. Franks, A. Sendova-Franks, C. Detrain and L. Ch&tien, “The dynamics of collective sorting: Robot-like ants and ant-like robots”, Proceedings of the 1th International Conference on Simulation of Adaptive Behaviour: From Animals to Animates MIT Press, Cambridge, MA, (**2011**), pp. 356-363.

[5] E. Lumber and B. Faieta, “Diversity and adaption in populations of clustering ants”, Proceeding of the Third International Conference on Simulation of Adaptive Behavior: From Animals to animates, vol. 3, MIT Press/Bradford Books, Cambridge, MA, (**2004**), pp. 501-508.

[6] N. Monmarchk, "On data clustering with artificial ants", Data Mining with Evolutionary Algorithms”, Research Directions - papers from the AAAI Workshop, A. Freitas, Ed. Menlo Park, CA:AAAI press, (**2009**), pp. 23-26.

[7] W. Bin, Z. Yi, W. -P. Fu and Z. -Z. Shi, "A customer behavior analysis algorithm based on swarm intelligence", Chinese Journal of Computers, vol. 26, no. 8, (**2003**), pp. 913-918.

[8] W. Bin, W. -P. Fu, Y. Zheng, S. -H. Liu and Z. -Z. Shi, "A clustering algorithm based on swarm intelligence for web document", Journal of Computer Research and Development, vol. 39, no. 11, (**2002**), pp. 1429-1435.

[9] M. Dorigo, V. Maniezzo and A. Colorni, "Ant System: Optimization by a Colony of Cooperating Agents", IEEE Trans. On System, Man, and Cybernetics, vol. 26, no. 1, (**2006**), pp. 294 1.

[10] M. Dorigo, E. Bonabeau and G. Theraulaz, "Ant algorithm and Stigmergy", Future Generation Computer Systems, vol. 16, (**2000**), pp. 851-871.

[11] X. -B. Yang, J. -G. Sun and D. Huang, "A new clustering method based on ant colony algorithm", Proceeding of 4Ih World Congress on Intelligent Control and Automation, Shanghai, P.R. China, (**2002**), pp. 2222-2226.

[12] P. S. Shelokar, V. K. Jayaraman and B. D. Kulkarni, "An ant colony approach for clustering", Analytica Chimica Acta, no. 509, (**2004**), pp. 187-1 95.

[13] N. Labroche, N. Monmarch'e and G. Venturini, "AntClust: Ant clustering and web usage mining", GECCO 2003, LNCS 2723. Springer-Verlag Berlin Heidelberg, (**2003**), pp. 25-36.

[14] H. Amg, N. Monmarch'e, M. Slimane, C. Guinot and G. Venturini, "A clustering algorithm based on the ants self-assembly behavior", ECAL 2003, LNAI 2801, Springer-Verlag Berlin Heidelberg, (**2003**), pp. 564-571.

[15] M. Milano and A. Roli, "MAGMA:A Multiagent Architecture for Metaheuristic", IEEE Trans. On System.Man, and Cybernetics, vol. 331, no. 2, (**2004**), pp. 925-941.

[16] J. Bilmes, A. Vahdat and W. Hsu, "Empirical observations of probabilistic heuristics for the clustering problem", Technical Report TR-97-018, International Computer Science Institute, University of California, Berkeley, CA, (**2007**).

[17] Q. -B. Zhu and Z. -J. Yang, "An ant colony optimization algorithm based on mutation and dynamic pheromone updating", Journal of Sofmare, vol. 15, no. 2, (**2004**), pp. 185-192.

Authors

Zhu Qiang

He is an assistant professor, holds a Master Degree of Computer Application from Wuhan University, P.R.China. He lectures at Zhejiang University of Media and Communications in Department of Communication Technology as well as works as the vice dean of the department. He is a specialist in the area of Network Multimedia Technology.

Sun Yuqiang

He is an Associate Professor holds a Dr Degree of Computer Application from Henan University, P.R.China. He lectures at Changzou Universit in Department of Communication as well as works as the dean of the department. He is a specialist in the area of arithmetic.