A Novel Method Based on Artificial Neural Network to Production Scheduling

Fuqing Zhao[1], Qiuyu Zhang[1], Dongmei Yu[1],Yahong Yang[2]

[1] School of Computer and Communication, Lanzhou University of Technology, 730050 Lanzhou, P.R.China
{zhaofq, zhangqy, yudm, xhchen}@mail2.lut.cn
[2] College of Civil Engineering, Lanzhou University of Techchnology, 730050 Lanzhou, P.R. China
{yangyahong}@mail2.lut.cn

Abstract

The job shop scheduling problem is one of the most general and difficult of all traditional scheduling problems. The goal of this research is to design and develop a job shop scheduling system (a scheduling software) that can generate effective job shop schedules using the multi-layered perceptron (MLP) networks. A method for organizing sample data has been developed using a de-notation bit to indicate processing sequence and processing time of a job simultaneously. The BP training process has been used to control local minimal solutions. Additionally, a heuristics has been suggested to improve and revise the initial production schedule. The proposed production activity schedule system is tested in a real production environment and simulation results show that the proposed model and algorithm are effective.
Keyword: Job shop scheduling, Multi-layered perceptron (MLP), Production planning

I. Introduction

Job-shop scheduling is one of the most basic models of scheduling theory [1]. A job shop consists of a set of m different machines $\mu = \{M_1, M_2, \cdots, M_m\}$ that perform operations of jobs [2]. There is a set of n jobs $\xi = \{J_1, J_2, \cdots, J_n\}$, where each job J_i has a specified processing order through the machines. That is, each J_i is composed of an ordered list of operations $(O_{i,1}, O_{i,2}, \cdots, O_{i,n_i})$, where $O_{i,j}$ is determined by the machine required, denoted by $\mu_{i,j} \in \mu$, and the duration of the operation, $d_{i,j}$. The rest of the assumptions are as follows:

(a) no machine can process more than one job (operation) at a time,
(b) the processing of the operations cannot be interrupted,
(c) all jobs and all machines are available from time 0 on.

Three approaches have traditionally been applied to solving job shop scheduling problems. They are priority rules, combinatorial optimization, and constraints analysis[3]. Each has its merits and weakness. The priority rules approach provides feasible scheduling solutions but they may not be optimal. The combinatorial optimization approach provides optimal solutions but may not have a solution for every scheduling problem, especially for the large-scale problems. The constraints analysis approach provides a set of feasible solutions that meet certain technical requirements for the scheduler to choose from[3][4].

In recent years, with the advancement of computer technology, both researchers and practitioners have turned to the knowledge-based problem solving approach to search for effective job shop scheduling methods [5][6][7][8]. The job-shop scheduling problem is an NP-complete problem and is usually very hard to find the optimal solution. An adaptive neutral network approach is able to provide feasible solutions through adapting its connection weights and biases of neutral units.

The research we conducted is based on the production activities of a manufacturing firm. Each job has its own product design and its technical information is very complicated. The product design information is often unavailable at the time the job should be dispatched and this situation causes delays in the firm's production planning and control. The firm does not have a standardized BOM for all its products and the computer-aided process planning (CAPP) software is not available [6][11]. Our objective is to design a production activity scheduling system (PAS) that can be used in a job shop manufacturing environment to improve production performance such as due dates management, lead time and reliability so as to improve the firm's overall profitability. We also consider the cost of hardware that the production activity scheduling software will require once the software is put into use. Considering both the scale of the network and the efficiency of the PAS system, we decide to employ the MLP networks to solve the day-to-day job shop scheduling problem.

The paper is organized as follows. Section 2 provides background information about the MLP and job shop scheduling problem(JSSP). Section 3 presents the neural network algorithm that is embedded in the PAS system and the solutions to some technical problems related to the system. An example case that utilizes the PAS software and the conclusions are provided in Sections 4 and 5 respectively.

II. Research Background

A. *The multi-layered perceptron neural network*

Artificial neural networks are alternative computation techniques that can be applied to solve production scheduling problems [3][9]. The MLP network is one of the most popular neural network architectures that fits a wide range of applications such as forecasting, process modeling, pattern discrimination and classification [12][13][14][15]. An MLP neural network may be trained on data generated from a real world problem or from a sophisticated model of the process. MLP can often provide outputs of adequate accuracy over a limited range of input conditions, with the advantage of requiring a lot less computation than other modeling methods [16][17].

MLP is trained by experimental data and the mapping is implemented as

$$f : S^n \to R^m$$

Once an activity sample is applied to the network, the system modifies the synaptic weights in accordance with the above mapping. After training, an MLP network behaves like an expert system. In real applications, the actual output of the network is the closest approximation of the outcome created by the set of data samples.

B. *Modeling and analysis of JSSP*

1. Notations for EJSSP

The symbols for modeling scheduling problems are as follows:

n ---number of jobs;

n_i --- number of operations of job i

m --- type number of various resources;

r_s --- number of resources of type s, $s \in [1,2,\cdots,m]$;

R_i ---set of pairs of operations $\{k,l\}$ belonging to job i, where operation k precedes operation l;

Q_i ---set of pairs of operations $\{k,l\}$ belonging to job i, for any operation k and operation l;
N_q ---set of operations requiring resource $q, q \in [1,2,\cdots,r]$;
H --- large enough positive number;
t_{il} ---processing time of operation l of job i, $l \in [l,\cdots,n_i]$;
x_{ik} ---starting time of operation k of job i, $k \in [1,\cdots,n_i]$;
x_{si} ---starting time of the first (or free) operation of job i;
x_{ie} --- completion time of the last (or free) operation of job i ;
a_i ---availability time of job i ;
d_i ---delivery due date of job i ;
$[i,k]$ --- the k th operation of job i, also called operation k in short if no confusion is caused;

$$y_{kl} = \begin{cases} 1 & \text{if operation } k \text{ precedes operation } l \\ 0 & \text{otherwise} \end{cases}$$

where $\{k,l\} \in Q_i, i = 1,2,\cdots,n$;

$$z_{ij} = \begin{cases} 1 & \text{if operation } i \text{ precedes operation } j \\ 0 & \text{otherwise} \end{cases}$$

where $\{i, j\} \in N_q, q = 1,2,\cdots,r_s, s = 1,2,\cdots,m.$

Note: $i \in [1,\cdots,n]$; $s \in [1,\cdots,m]$; free operation means operation without precedence restrictions from technological planning.

2. Expanded job-shop scheduling problem(EJSSP)

EJSSP is a deterministic and static scheduling problem. There are m distinct machines to process n jobs that have their specific processing routines. Each job's operation has its precedence and takes up a deterministic time period at a specific machine. At one time, there is only one operation at a machine and the job does not leave this machine until the operation is completed. It can be seen easily that EJSSP is significantly more general than the standard JSSP[18][19].

The objectives considered are:

1) to minimize the end time of the last completed job, or
2) to minimize the total penalty for tardy and early jobs.

Scheduling problems with the following two objectives are tested in this paper although the hybrid scheduling approach is not restricted to a specific objective.

Minimizing the completion time of the last completed job,

$$Min\ Z = \max_i (x_{ie} + t_{ie}) \qquad (1)$$

Minimizing the total penalty for early and tardy jobs,

$$Min\ Z = \sum_{i=1}^{n} \sum_{t=1}^{d_i} \sum_{k=1}^{n_i} [z_{ik}(t) \times \max(0, d_i - x_{ie}) + y_{ik}(t) \times \max(0, x_{ie} - d_i)] \qquad (2)$$

where h_i and w_i are the early and tardy penalty weights for job i. It should be noted that if Eq. (2) were served as scheduling objective, constraint equation (7) could be omitted, i.e. d_i should be set as a positive infinite number.

A feasible solution means that the scheduling satisfies all constraint conditions. There are three types of major constraints for any operation as follows:

1) Precedence constraint. Precedence constraint means that some jobs must be processed at different machines in fixed precedence sequence defined by technological planning. Concretely, the l th operation of job i must be before the k th operation of the same job, if $\{k,l\} \in R_i$, i.e.

$$\sum_{i=1}^{n}\sum_{l=1}^{n_i} x_{il} - \sum_{i=1}^{n}\sum_{k=1}^{n_i} x_{ik} \geq \max\{\sum_{j=1}^{m}\sum_{t=1}^{d_i}(t+d_{ij})y_{ij}(t), \sum_{j=1}^{m}\sum_{t=1}^{d} ty_{kj}(t)x_{il} - t_{ik}\} \qquad (3)$$

2)Resource constraint. Resource constraint means that any resource can only provide service for one operation at a time; for example, resource q can only select one job to serve among jobs waiting to be processed in the queue at any time.

3) Job (hidden) constraint. Although there may be no precedence constraint among some operations of a job, the constraint that the operations l and k could not be processed at the same time still exists because these two operations were done at the same job, i.e.

$$x_{jl} - x_{ik} - t_{ik} + H(1 - z_{kl}) \geq 0 \quad if\{k,l\} \in N_q \qquad (4)$$

where $z_{kl} = 0\ or\ 1$.

It is indicated that $\{k,l\} \in R_i$, an operation of job i, must satisfy Eq. (1) and $\{k,l\} \in Q_i$, another operation of job i, must satisfy Eq. (4).

4) Starting and completion time constraint. In practice, the starting time and the completion time of a job are restricted by the job available time and the due date of delivery. Mathematically, it can be depicted by Eqs. (5).

$$\sum_{k=1}^{n}\sum_{t=1}^{d}\{x_{si}(y_{ik}(t) - z_{ik}(t)) - a_i\} \geq 0 \quad i \in [1,\cdots,n] \qquad (5)$$

III. Developing a production activity scheduling system using MLP

A. The relevant Parameters

The relevant parameters used in the PAS algorithm are as follows:

η :learning rate, $\eta > 0$

β : momentum parameter, $0 < \beta < 1$

α : oscillation parameter, $0 < \alpha < 1$

ϖ : matrix of weight values

$\varpi(t)$: weight value between neurons after the t th change

$\Delta\varpi$:weight change

ML : memory length

y : ideal target output of object tier layer

y_i : ideal target output of unit i

$\hat{y}$: real target output

$\hat{y}_i$: real target output of unit i

z : real output of hidden layer

z_i : real output of unit i in hidden layer

n : the number of neurons in the input layer

h : the number of neurons in the hidden layer

m : the number of neurons in the output layer

s : training sample assemble

λ : real constant

$\psi(*)$: monotonically increasing real function that is independent of the mapping $f(*)$ which it approaches
ε : a positive constant
$g_i(*)$: real continuous function

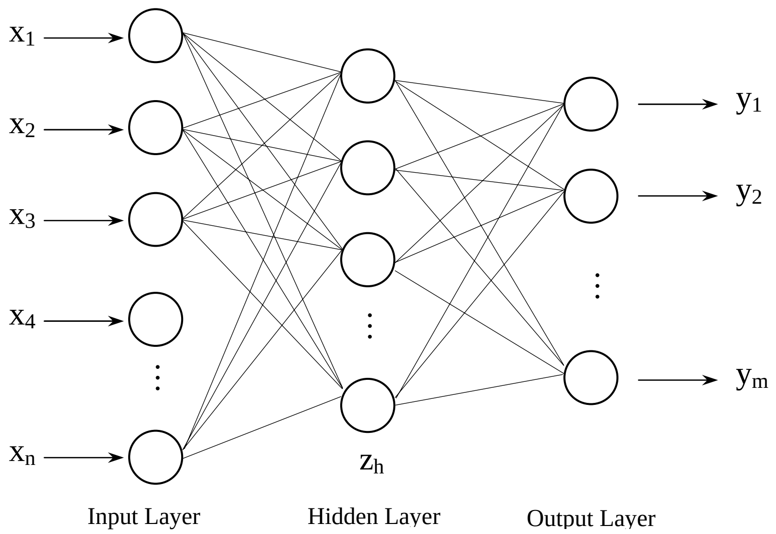

Fig. 1. Three-layer MLP neural network.

B. System structure

The MLP has an input layer, a hidden layer, and an output layer to implement the mapping $y = f(x)$. The system structure is shown in Fig. 1.

In this structure, the input units transfer each component of the input vector x to the computing units in the hidden layer. The disposal units in the hidden layer implement the following input/output relation:

$$Z_k = \sum_{j=1}^{n} \lambda^k \psi(x_j + k\varepsilon) + k, \quad k = 1,2,\cdots,h. \qquad (6)$$

where W_{kj} is the synaptic weight between the hidden unit k and the input unit j, θ_k is the value of disposal unit k.

The m output units are provided with the following input/output function:

$$\hat{y}_i = \sum_{k=1}^{h} g_i(z_k), \quad i = 1,2,\cdots,m \qquad (7)$$

generally $g_i(z_k) = W_{ik} Z_k$.

(1) Learning algorithm

The error function is

$$e(w) = \tfrac{1}{2}\sum_{k=1}^{s} \left\| f(xk) - \hat{y}_k \right\|^2 \qquad (8)$$

For the output layer

$$\frac{\partial e(\varpi)}{\partial \varpi_{ik}(t)} = \eta \sum [(y_n^r - \hat{y}_i^r) z_k^r] \;\; i = 1,2,\cdots,m; k = 1,2,\cdots,h \qquad (9)$$

where r is the sequence number of the sample in the sample assemble, y_i^r the output of component i of the ideal value of the sample, $\hat{y}_i^r$ the output of component i of the actual value of sample r, and y_k^r the output of component k of the actual hidden layer value of sample r.

For the hidden layer

$$\frac{\partial e(\varpi)}{\partial \varpi_{kj}(t)} = \eta \sum_{r=1}^{s} \left\{ \sum_{i=1}^{m} [(y_i^r - \hat{y}_i^r)\varpi_{ik}(t)] z_k^r (1 - z_k^r) x_j^r \right\} \qquad (10)$$

(2) Strategy for adjusting the weights

When $t^* \in [t - ML, t], t \geq ML \ or \ t^* \in [1, t], t < ML$

$$\varpi(t+1) = \varpi(t) - \eta \frac{\partial e(\varpi)}{\partial \varpi(t)} + \beta[\varpi(t) - \varpi(t-1)] = \varpi(t) + \Delta\varpi(t) \qquad (11)$$

where α is a random number between (0,1).

(3) Modifying results

(a) If the training result satisfies the PAS modeling requirement, the current neural network then can be adopted as the final production activity-scheduling model.

(b) If the training result does not match the requirement, and it has passed over the predetermined maximal training time limit, then,

1) If among the minimum queue M there exists a minimum with an error equal to or smaller than the predetermined error lower limit, then the current neural network can be used as the PAS model;

2) otherwise, this training process has to be treated as a failed training.

IV. Experiments analysis

The proposed PAS system has been implemented in a real production environment. The results are promising. Here we present an example case to demonstrate the actual implementation of the PAS system that has been developed in this study.

The simulation example is a $6/6/J/C\max$ production activity scheduling problem by using MLP. The number of jobs is six $\{J_1, J_2, J_3, J_4, J_5, J_6\}$; the number of machines is six $\{1,2,3,4,5,6\}$; the processing time is day. The total number of samples is 300, where the trained neural network—MLP—is structured with the number of input units equal to 12, the number of hidden units is six, and the number of output units is six. The network structure is shown in Fig. 1.

Table 1 gives the processing order and processing time information of each job.

According to the processing order and processing time of each job in Table 1, the sample can be interpreted as follows: X $= (X_1, X_2, \cdots, X_{12})$, where X_1 = 123456, X_2 = 332421. X_1 shows that the processing order of job 1 is from machine 1–2, then 3, 4, 5 and 6. The same rule applies to $X_2 - X_{12}$.

A large number of X samples are collected to train the MLP. When the training is done, the MLP is applied to a new scheduling problem of the same type. For this 6/6/J/Cmax problem the solution is as follows:

$Y = (Y_1, Y_2, Y_3, Y_4, Y_5, Y_6)$ = (152364,421653,621345,531246,365421,243516).

where the term Y_1 = 152364 means that machine 1 is firstly occupied by job 1, then jobs 5, 2, 3, 6, 4 (see Table 2).

Table 1. processing order and processing time information of each job

Jobs	Process order and Process time					
J1	1(3)	2(3)	3(2)	4(4)	5(2)	6(1)
J2	2(3)	3(1)	1(4)	4(2)	6(3)	5(2)

J3	5(4)	4(3)	1(1)	6(2)	3(2)	2(1)
J4	2(2)	3(4)	6(4)	4(3)	5(1)	1(3)
J5	4(5)	1(2)	5(1)	6(5)	2(3)	3(4)
J6	3(1)	6(2)	5(4)	2(2)	1(6)	4(1)

Table 2. Data for scheduling Gantt chart

Machine	Date	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
1		1	1	1			5	5	2	2	2	2	3	6	6	6	6	6	6	4	4	4
2		4	4	2	2	2	1	1	1	6	6	5	5		3							
3		6					2			1	1			3	3	4	4	4	5	5	5	5
4		5	5	5	5	5	3	3	3			1	1	1	1		2	2	4	6		
5		3	3	3	3	6	6	6	6	5	4	4	4	4	2	2	1	1				
6		2	2					4	4	3	3	3				5	5	5	1	1	6	6

A. System structure

In this case, the MLP has the following structure,

Table 3. Parameters used in the MLP

Maximum training times	20000		
Number of samples	300	Memory length	80
Impulse modulus	0.85	Network	12-6-6
Learning modulus	0.1	Training	980
Minimal error	5E-6	Training	10.2 s

B. The training convergent curve

The training convergent curve of the whole training process is shown in Fig. 2. According to the curve of training error, we can see that oscillation emerges at the beginning of the training process. The oscillation reflects the process of the solution trapped into and sprung out of the local minimum. The revised system output data are acceptable in the actual production scheduling process.

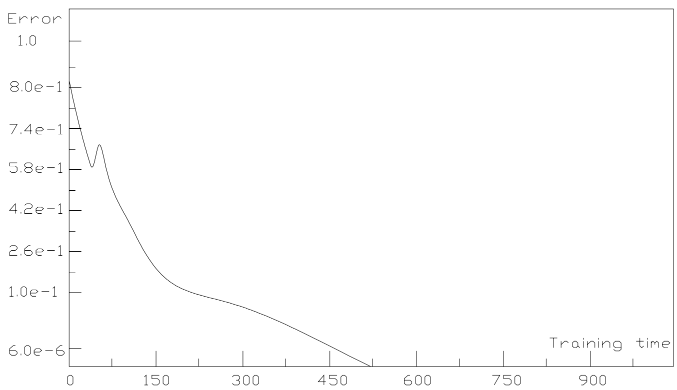

Fig. 2. The training convergent curve

C. Revision

As the MLP for the PAS system is employed, a deviation of the outputs from the expected outputs is not unusual. For example, the ideal output of the example case is

$Y = (Y_1, Y_2, Y_3, Y_4, Y_5, Y_6)$ = (543612,136425,564123,126543,635412,251364)
and the computation output obtained is
$Y^* = (Y_1^*, Y_2^*, Y_3^*, Y_4^*, Y_5^*, Y_6^*) =$ (543594,136423,564118,126574,635412,251364).

The errors are obvious, for example, Y_1^* = 543594, where job 5 appears twice and there is no 9th job. then We modify Y^* by applying the following heuristic rules:

1. Revisions should be taken orderly from Y_1 to Y_n.
2. If a repetition appears, replace the repeated number by the closest, unused number. In Y_1^* = 543594,the numbers near 5 are 4 and 6, but 4 is already used in the 2nd place, so it can only be replaced by 6, then Y_1^* = 543694. In the same manner, 1 should replace 9 because there are only 6 jobs. In the same consideration, the number 4 which is in the 6th place can only be replaced by 2. After revising the initial output, we get Y_1 = 543612. Using the same approach to revise the whole set, the PAS system generates the scheduling solution

$Y = (Y_1, Y_2, Y_3, Y_4, Y_5, Y_6)$ = (543612,136425,564123,126543,635412, 251364).

V. Conclusion

In this research, we have designed, developed, and implemented a PAS using the MLP neural networks. A method for organizing sample data has been developed using a denotation bit to indicate processing sequence and processing time of a job simultaneously. The BP training process has been used to control local minimal solutions. Additionally, a heuristics has been suggested to improve and revise the initial production schedule. The proposed production activity schedule system is tested in a real production environment and proven to be useful and feasible. The PAS system helped the manufacturing firm that we studied improve their production activity control, customer service level, and profitability.

Acknowledgements

This research is supported by National Natural Science Foundation of GANSU province (contract No. 3ZS042-B25-005) and Natural Science foundation of GANSU province(contract No. ZS032-B25-008).

References

[1] Aydin, M. Emin, Fogarty, Terence C., “A simulated annealing algorithm for multi-agent systems: A job-shop scheduling application”. Journal of Intelligent Manufacturing, 15(4): 805-814,2004.

[2] Smith, J.S., Peters, B.A., Srinivasan, A., “Job shop scheduling considering material handling”. International Journal of Production Research,37(7) :1541-1560,1999.

[3] Yu, H., Liang, W., “Neural network and genetic algorithm-based hybrid approach to expanded job-shop scheduling”. Computers and Industrial Engineering,39(3-4): 337-356,2001.

[4] Jain, A.S., Meeran, S., “Deterministic job-shop scheduling: Past, present and future”. European Journal of Operational Research ,113(2): 390-434,1999.

[5] Nagaraj, P.; Selladurai, V. , “Analysis of optimum batch size in multistage, multifacility and multiproduct manufacturing systems”. International Journal of Advanced Manufacturing Technology,19(2):117-124,2002.

[6] Lee, D.H., Kiritsis, D., Xirouchakis, P., “Iterative approach to operation selection and sequencing in process planning”. International Journal of Production Research, 42(22):4745-4766,2004.
[7] Tang, Ou, Grubbstrom, Robert W., “Rescheduling considerations for production planning using control theory”. International Journal of Industrial Engineering : Theory Applications and Practice ,9(3) :258–264 ,2002.
[8] Anglani, A., Grieco, A., Pacella, M., Tolio, T., “Object-oriented modeling and simulation of flexible manufacturing systems: A rule-based procedure”. Simulation Modeling Practice and Theory, 10(3-4): 209–234,2002.
[9] Fonseca, Daniel J., Navaresse, Daniel. , “Artificial neural networks for job shop simulation. Advanced Engineering Informatics” , 16(4) :241-246,2002.
[10] El-Bouri, Ahmed, Balakrishnan, Subramaniam., Popplewell, Neil., “Sequencing jobs on a single machine: A neural network approach”. European Journal of Operational Research ,126(3): 474–490,2000 .
[11] Zhou, Xionghui,Wang, Yu, Lou, Zhenliang,etal., “Study on agile manufacturing technology and its system development for die and mold”. Chinese Journal of Mechanical Engineering 11(6):59-64,2004.
[12] Lin, Daw-Tung., “Target components discrimination using adaptive time-delay neural network”. Journal of Information Science and Engineering,20(5): 959–980 ,2004.
[13] Taylor, James W.,Buizza, Roberto., “Neural network load forecasting with weather ensemble predictions”. IEEE Transactions on Power Systems ,17(3): 626–632,2002 .
[14] Huang, H.C., Hwang, R.C., Hsieh, J.G., “Short-term power load forecasting by non-fixed neural network model with fuzzy BP learning algorithm”. International Journal of Power and Energy Systems,22(1):50-57,2002.
[15] Kim, Byungwhan, Kim, Sebum L., “Partial diagnostic data to plasma etch modeling using neural network”. Microelectronic Engineering, 75(4): 397–404,2004.
[16] liao, Gwo-Ching,Tsao, Ta-Peng., “Application of fuzzy neural networks and artificial intelligence for load forecasting”. Electric Power Systems Research. 70(3): 237-244,2004.
[17] Satish, B.,Swarup, K.S., Srinivas, S. etal., “Effect of temperature on short term load forecasting using an integrated ANN”.Electric Power Systems Research,72(1): 95-101,2004.
[18] Ho, N.B., Tay, J.C. GENACE, “An efficient cultural algorithm for solving the flexible job-shop problem”. Proceedings of the 2004 Congress on Evolutionary Computation, CEC2004: 1759-1766.
[19] Torres, Philippe,Lopez, Pierre., “On not-first/not-last conditions in disjunctive scheduling”. European Journal of Operational Research ,127(2) :332–343,2000.

Fuqing Zhao is a PhD candidate in School of Computer and Communication Engineering at the Lanzhou University of Technology. He is currently working as a research assistant at the Key Laboratory of Computer Science of Institute of Software in Chinese Academy of Science. His research interests include CAD/CAM/CAPP,collaborative engineering, design theory and methodology. Mr Zhao received his BASc in Mechanical engineering in 1998 and MASc in 2001 from Lanzhou University of Technology.

Qiuyu Zhang is the programme Director and Associate Professor in the School of Computer and Communication Engineering, Lanzhou University of Technology. His research interests include: Software engineering; Grid Computing; CAD/CAM/CAPP; Computer Intelligence, particularly in the area of Computer Intelligence in application use.

Dongmei Yu is the Professor in the School of Computer and Communication Engineering, Lanzhou University of Technology. The research interests of professor Yu include: Software engineering; Manufacturing Technology; CAD/CAM/CAPP; ERP; Computer Intelligence, particularly in the area of Computer Intelligence in application use.

Yahong Yang is the MASc candidate of College of Civil Engineering, Lanzhou University of Technology. She received BSc in Environment Engineering from the Haerbin Engineering University in 2001. Her research interests include: Computer technology in sewage disposal; Computer Intelligence in sewage disposal; Neural Network; Genetic algorithm.