Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Quantum Genetic Based Scheduling Algorithm for stochastic flow shop scheduling problem with random breakdown

Jinwei Gu*. Xingsheng Gu**. Bin Jiao***

**Research Institute of Automation, East China University of Science and Technology, 200237, Shanghai, China(e-mail: gujinwei1982@163.com)*
***Research Institute of Automation , East China University of Science and Technology,200237,Shanghai,China(e-mail: xsgu@ecust.edu.cn)}*
****Electrical Engineering Department, 200237,Shanghai ,China(e-mail:binjiao@163.com)*

Abstract: A Quantum Genetic Based Scheduling Algorithm (QGBSA) for stochastic flow shop scheduling with random breakdown and random repair time is proposed in this paper, which combines stochastic programming and stochastic simulation theory, quantum compute and genetic algorithm together. In the QGBSA, the Q-bit based representation in discrete 0-1 hyperspace is employed, which is then converted into decimal scheduling code, and quantum gate is used to update the current generation. Meanwhile, catastrophe operator is added to avoid premature. In order to improve the effectiveness of scheduling scheme, the stochastic programming theory is used to set up a stochastic flow shop scheduling model without breakdown. Then we consider two different working modes—preemptive-resume and preemptive-repeat under breakdown. The former processes the rest part of interrupted job while the later reprocesses the interrupted job. Under each working mode, the situations where breakdown happened at any time or at machine's life-span time were discussed. Finally, compared with traditional Genetic Algorithm (GA), computational results show the feasibility and effectiveness of QGBSA policy.

1. INTRODUCTION

Flow shop scheduling problem with strong engineering background is an important optimization problem, which is of the difficulties such as inaccurate estimation of optimization objective and NP-hardness. Meanwhile, there are many uncertainties in process industry such as machine breakdown, operator–stock condition, changes in availability date and latest completion times; we must consider them to ensure the production running successfully. See (Xingsheng Gu, 2000). Commonly, the processing time of jobs are known in advance only with uncertainty. At present, basic methods for solving scheduling problem with uncertain processing time are stochastic programming, fuzzy programming, rough sets, etc. If the processing time of each job is initially described in terms of a probability distribution, then we call such problem as stochastic scheduling, which could be solved by stochastic programming. Moreover, the machines are not always available and breakdowns can happen at any time. So considering random breakdowns is more realistic. Generally, the machine's repairing time is fixed, but here we let it obey random distributions.

This stochastic model is much more realistic than deterministic models while expressing real world problems. By means of the stochastic model, it becomes much easier to meet the demands of large projects in the face of uncertainty in the initial project parameters. Unfortunately, stochastic scheduling problems tend to be far more difficult to deal with from both a theoretical and computational point of view than their deterministic counterparts. So we use stochastic programming to set up stochastic model, and employ stochastic simulation to achieve stochastic sampling.

In the literature research of stochastic flow shop scheduling problem with breakdowns, many different approaches have been applied and a rich harvest have been obtained, such as heuristic arithmetic, Genetic Algorithm, etc, see (Allaoui, H. and Artiba, A. 2004; Li, Su-Fen et al., 2005) . A detailed review of scheduling in a flow shop with breakdowns is given by M. Gourgand et al. (2000). Meanwhile, ALLAHVERDI A, et al. (1995) also give some research results concerning flow shop with stochastic breakdown. Besides, ALCAIDE D et al. (2002) propose an approach to converts breakdowns scheduling problems into a series of problems without breakdowns.

Recently, quantum evolution, burgeon of computing theory, comes into being, which is also a combined product of quantum physics and computer science. As we all known, evolution algorithm simulates heuristic searching based on nature species' evolution. Quantum evolution algorithm, combing quantum system's adding character and parallel character, could improve the efficiency of algorithm. So far, many efforts on quantum computer have been paid due to its superiority to classical computer for various specialized problems. As a novel evolutionary method, Quantum evolution Algorithm has gained much attention and application for both function and combinatorial problems, but there is little research on stochastic scheduling.

2. PROBLEM DESCRIPTION

The permutation flow shop scheduling problem is often designated by the symbols $n/m/P/C_{\max}$, where a set of jobs $Job=\{1,...,n\}$ have to be processed on m machines in the same order. The processing of each job on each machine is an operation which requires the exclusive use of the machine for an uninterrupted duration called the processing time. Here, we assume that the jobs are all available for processing at time zero and, at any time, each job can be processed on at most one of the machines which are subjected to stochastic breakdowns. On the other hand, each machine can process at most one job at a time.

In this paper, we assume S is the job permutation $S=\{s_1,s_2,...,s_n\}$. The processing time of job i on machine j is a nonnegative random p_{ij} with a mean μ_{ij} and a variance σ_{ij}^2, and all the processing times are independent and following normal distributions. We also suppose each machine happen breakdown only once. The breakdown time point b_i and repair time length r_i on machine i obey normal distributions and their means are $b\mu_i$, $r\mu_i$ and the variances are $b\sigma_i$, $r\sigma_i$.

$\mu_{ij}=E\{p_{ij}\},\sigma_{ij}=Var\{p_{ij}\}$

$b\mu_i=E\{b_i\},b\sigma_i=Var\{b_i\}$

$r\mu_i=E\{r_i\},r\sigma_i=Var\{r_i\}$

p_{ij} —the processing time of job j on machine i, which is an independent stochastic variable

m_i —continuous available time on machine i

st_{ij} —starting time of job j on machine i, a stochastic variable

c_{ij} —completing time of job j-th on machine i, a stochastic variable.

3. STOCHASTIC FLOW SHOP WITH RANDOM BREAKDOWN MODEL AND SOLVING METHOD

Only considering the random processing time, the expected completing times of each job on each machine are described as follows (Expected value model):

$$E\{st_{1s_1}\}=0 \quad (1)$$

$$E\{c_{1s_1}\}=E\{st_{1s_1}\}+E\{p_{1s_1}\} \quad (2)$$

$$E\{c_{1s_j}\}=E\{c_{1s_{j-1}}\}+E\{p_{1s_j}\}\ j=2,...,n \quad (3)$$

$$E\{c_{is_1}\}=E\{c_{(i-1)s_1}\}+E\{p_{is_1}\}\ i=2,...,m \quad (4)$$

$$E\{c_{is_j}\}=\max\{E\{c_{is_{j-1}}\},E\{c_{(i-1)s_j}\}\}+E\{p_{(i-1)s_j}\}$$
$$i=2,...,n$$
$$j=2,...,m \quad (5)$$

When the stochastic breakdown is added into the problem, the solving method can be described as follows.

Woking mode 1 (preemptive-resume model): After repairing, the rest part of job being interrupted by breakdown will be processed.

Woking mode 1 contains two cases as follows.

Case 1: Breakdown can happen at any time.

The expected time when breakdown occurs on machine i is:

$$E\{b_i\}=E\{st_{is_1}\}+E\{m_i\}\ \ i=1,2,...,m$$

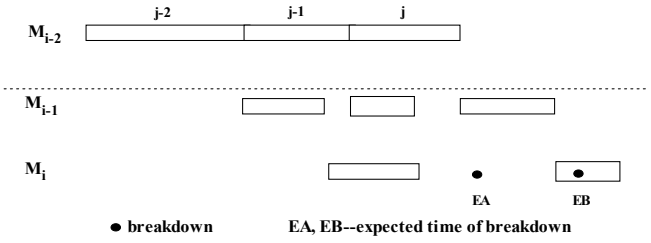

Fig. 1. The gantt chart of flow shop scheduling problem with breakdown

EA is the time when breakdown happened during machine's idle time interval. EB is the breakdown time happened in machine's processing time interval, which can be shown in Fig 1.

The expected completing time of job j can be calculated as follow:

$$If\ \ \left(E\{c_{is_{j-1}}\}\le EA\right)and\left(E\{c_{(i-1)s_j}\}>EA\right)$$
$$E\{c_{is_j}\}=\max\{E\{st_{is_j}\},EA+E\{r_i\}\}+E\{p_{is_j}\}$$
$$Elseif\ \ \left(E\{st_{is_j}\}\le EB\right)and\left(\left(E\{st_{is_j}\}+E\{p_{is_j}\}\right)>EB\right)$$
$$E\{c_{is_j}\}=E\{st_{is_j}\}+E\{p_{is_j}\}+E\{r_i\}$$
$$end\ if$$

Case 2: Each machine has its life-span, if the total working time of a machine exceeds its life-span, breakdown will happen.

In this case, breakdown only happens at EB time. So the expected value of breakdown time on machine i is its life-span time.

The expected completing time of job j can be calculated as following:

Without loss of generality, we suppose EB to be the breakdown of machine i, and denote by Q the last job before EB.

$$Let\ \ E\{M_i\}=\sum_{k=Q+1}^{Q+j-1}E\left\{p_{is_k}\right\},E\{N_i\}=\sum_{k=Q+1}^{Q+j}E\left\{p_{is_k}\right\}\ \ i=1,2,...m$$

$$If\quad \left(E\{M_i\}\le EB\right)and\left(E\{N_i\}>EB\right)$$

$$E\left\{c_{is_j}\right\}=E\left\{st_{is_j}\right\}+E\left\{p_{is_j}\right\}+E\{r_i\}$$

end if

Woking mode 2 (preemptive-repeat model): After repairing, the job being interrupted by breakdown will be reprocessed.

In the same way, breakdown may happen at time EA or time EB. The expected completing time of job j can be calculated as follows:

Case 3: Breakdown can happen at any time.

$$If\quad \left(E\left\{c_{is_{j-1}}\right\}\le EA\right)and\left(E\left\{c_{(i-1)s_j}\right\}>EA\right)$$

$$E\left\{c_{is_j}\right\}=\max\left\{E\left\{st_{is_j}\right\},EA+E\{r_i\}\right\}+E\left\{p_{is_j}\right\}$$

$$Elseif\quad \left(E\left\{st_{is_j}\right\}\le EB\right)and\left(\left(E\left\{st_{is_j}\right\}+E\left\{p_{is_j}\right\}\right)>EB\right)$$

$$E\left\{c_{is_j}\right\}=EB+E\left\{p_{is_j}\right\}+E\{r_i\}$$

end if

Case 4: If the total working time of a machine exceeds its life-span, breakdown will happen.

$$Let\ \ E\{M_i\}=\sum_{k=Q+1}^{Q+j-1}E\left\{p_{is_k}\right\},E\{N_i\}=\sum_{k=Q+1}^{Q+j}E\left\{p_{is_k}\right\},i=1,2,...m$$

$$If\quad \left(E\{M_i\}\le EB\right)and\left(E\{N_i\}>EB\right)$$

$$E\left\{c_{is_j}\right\}=EB+E\left\{p_{is_j}\right\}+E\{r_i\}$$

end if

4. QUANTUM GENETIC BASED SCHEDULING ALGORITHM

In the QGBSA, Q-bit based representation is employed for exploration in discrete 0-1 hyperspace by using updating operator of quantum gate as well as genetic operators of Q-bit. Meanwhile, the Q-bit representation is converted to random key representation, which is then transferred to job permutation for objective evaluation (Li, Bin-Bin et al., 2007).

4.1 Basic quantum compute

Before we introduced the QGBSA, let's briefly review the basic knowledge of quantum compute.

Quantum compute uses quantum bit as information unit. The advantage of quantum bit is when there are two quantum bit, it can stay in adding states of four state 00,01,10,11.

The state of a Q-bit can be represented as follows:

$$\left|\psi\right\rangle=\alpha\left|0\right\rangle+\beta\left|1\right\rangle \qquad (6)$$

, where α and β are complex numbers that specify the probability amplitudes of the corresponding states.

Thus, $\left|\alpha\right|^2$ and $\left|\beta\right|^2$ denote the probabilities that the Q-bit will be found in the "0"state and "1"state respectively. Because our paper discusses flow shop scheduling problem, so α and β is in real number.

A Q-bit individual p, a string of m Q-bits, is defined as follows:

$$\begin{bmatrix}\alpha_1 & \alpha_2 & ... & \alpha_m\\ \beta_1 & \beta_2 & ... & \beta_m\end{bmatrix}$$

Where $\left|\alpha\right|^2+\left|\beta\right|^2=1, i=1,2,...,m$

4.2 Coding and Decoding of QGBSA

In QGBSA, the quantum unit is converted to random key representation, which is the final flow shop code. The detailed process can be described as:

Firstly, let η be a random number generated between $[0,1]$. If α_i of individual p satisfies $\left|\alpha_i\right|^2>\eta, i=1,...,m$, then set $r_i=1$, otherwise $r_i=0$.

Then, the binary representation is viewed as random key representation.

Finally, job permutation is constructed based on random key.

For example, consider a 3-job, 3-machine problem, let 3 Q-bits represent a job. Suppose a binary representation is [0 1 1| 1 0 1| 1 0 1] which is converted from Q-bit representation, then the random key representation is [3 5 5]. If values of two random key representations are different, let smaller random key denote the job with smaller index; otherwise, we let the first one denote the job with smaller index. So, the above random key representation is corresponding to job permutation [1 2 3]. Obviously, if enough Q-bits are used to represent a job, any job permutation would be constructed with the above strategy from binary representation based space. Example can be seen from Fig 2.

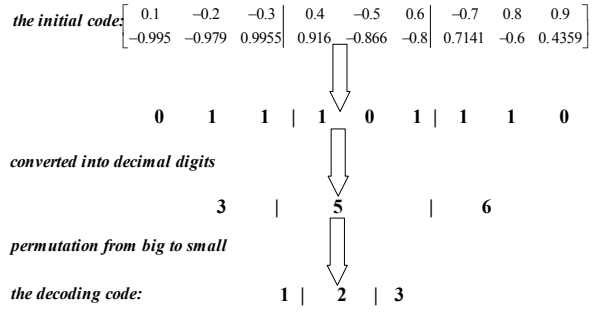

Fig. 2. Decoding way of flow shop

4.3 Selection of QGBSA

In this paper, the selection method of our QGBSA is gambling. Since many results concerning gambling selection could be found, here, details about it are omitted.

4.4 Crossover of QGBSA

Here, two-point crossover is used, which sets two crossover points randomly, and selects the section between the points from one parent and other sections outside the points from the other parent and recombines them.

4.5 Mutation of QGBSA

One position is randomly chosen, and then the corresponding α_i and β_i are exchanged. To avoid premature, a catastrophe operator is used. In this paper, if a solution does not change in certain consecutive generations, and it could be regarded to be trapped in local optimal, then the local best solution is reserved while the others will be replaced by solutions generated randomly.

4.6 Rotation Operator

The rotation gate is applied on the probability amplitude of quantum states in Quantum evolution algorithm in order to maintain diversity of population, which is an important updating method of quantum evolution algorithm.

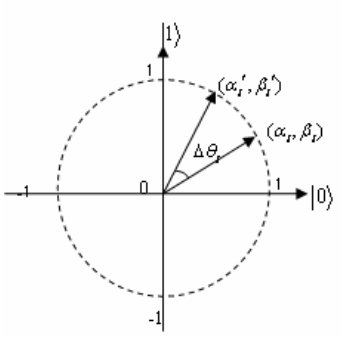

Fig. 3. Polar plot of the rotation gate for Q-bit individuals

The quantum transforming matrix accomplishes quantum mutation. The selection of quantum gate has many types: not gate, controlled Not gate, hadamard gate, rotation gate, etc. Here we choose rotation gate as quantum mutation. The plot of polar coordinate concerning quantum rotation gate can be described in Fig 3.

A rotation gate $U(\theta)$ is employed in quantum evolution algorithm to update a Q-bit individual as a variation operator. (α_i, β_i) of the i-th Q-bit is updated as:

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = U(\theta_i)\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \cdot \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix} \quad (7)$$

, where θ_i is rotation angle. Let $\theta_i = s(\alpha_i, \beta_i)\Delta\theta_i$, where $s(\alpha_i, \beta_i)$ is the sign of θ_i that determines the direction, $\Delta\theta_i$ is the magnitude of rotation angle whose lookup table is shown in Table 1. In the table, BES_i and BIS_i are the i-th bits of the best solution b and the binary solution r respectively.

Table 1. Rotation angle

BIS_i	BES_i	$f(r)<f(b)$	$\Delta\theta_i$	$\alpha_i\beta_i>0$	$\alpha_i\beta_i<0$	$\alpha_i=0$	$\beta_i=0$
0	0	False	0	0	0	0	0
0	0	True	0	0	0	0	0
0	1	False	0	0	0	0	0
0	1	True	0.05pi	-1	+1	1or-1	0
1	0	False	0.01pi	-1	+1	1or-1	0
1	0	True	0.025pi	+1	-1	0	1or-1
1	1	False	0.005pi	+1	-1	0	1or-1
1	1	true	0.025pi	+1	-1	0	1or-1

4.7 Procedure of QGBSA

Procedure of the QGBSA algorithm for stochastic flow shop scheduling with random breakdown is following:

Step 0: input initial parameters: PS (population size), Pc (crossover probability), Pm (mutation probability), Sampling Times, iterative generation (GN), let k=0;

Step 1: randomly generate an initial population $P_Q(t) = \{P_1^t, ..., P_N^t\}$, which was converted to a binary representation, then decoded into random key representation, which is corresponding to job permutation in decimal code.

Step 2: using stochastic simulation to calculate makespan of $P_Q(t)$, and then record the best scheduling result.

Step 3: if stopping condition (k=GN) is satisfied, then output the best scheduling result; otherwise go to step 4.

Step 4: perform quantum selection, crossover, mutation for $P_Q(t)$ to generate $P_Q'(t)$.

Step 5: if catastrophe condition is satisfied, perform catastrophe for $P_Q'(t)$ to generate $P_Q(t+1)$ and go to step 7; otherwise go to step 6.

Step 6: applying rotation gate $U(\theta)$ to update $P_Q'(t)$ to generate $P_Q(t+1)$.

Step 7: The generation $P_Q(t+1)$ was converted to decimal codes, which denote flow shop permutation. Then use stochastic simulation to calculate makespan of $P_Q(t+1)$, and update the best scheduling solution if possible. Let k=k+1, go back to step 3.

5. EXPERIMENT

The new algorithm is tested by solving benchmark Taillard problems. Five representative instances of different size have been selected, which are 20×5, 20×10, 20×20, 50×5, 50×10 .The results obtained are comparable to the GA method and show the appropriateness of the new algorithm on stochastic flow shop with random breakdown.

The processing time, breakdown time point and repairing time obey normal distributions. We use benchmark datas as processing times' mean value and variances are fetched from $[0,1]$. The mean value of breakdown time point is $b\mu_i = 100$, variance is $b\sigma_i = 10$. The mean value of repair time is $r\mu_i = 100$, variance is $r\sigma_i = 10$. Stochastic simulation is used to achieve stochastic sampling. The procedure is running on Pentium(R) 2CPU 3.06GHZ.

In QGBSA, the population size is 50, crossover probability is 0.9 and mutation probability is 0.1. If the catastrophe condition is satisfied, mutation probability is added another 0.5. Times of Iteration are 200 and stochastic time's sampling times is 20. In GA, the population size is 50, crossover probability is 0.9 and mutation probability is 0.2.

We take QGBSA to be compared with GA, using case1 as an example. First stochastic sampling technology is used to get a series solutions each time, from which, we find out those solutions with the min, average and max makespan. After ten runs on each problem instance, we get ten groups data. Then we calculate the average number of min, average and max makespan. The statistical results are shown in Table 2, where we can see QGBSA-based on stochastic flow shop scheduling problem with random breakdown can get much better solutions than GA with relatively less iterative times. The results obtained by QGBSA are very close to the best solution of the determinate FSSP. The average performance of the QGBSA is also better than that of the GA, especially for the large-scale problems. If the convergence generation is very small, GA is easy to trap in local optimal solution, but QGBSA won't have this phenomenon.

Fig. 3 is convergence curves of QGBSA and GA with different size. By comparison, the simulation solution shows clearly that the convergence rate of QGBSA is faster than the GA. Fig.4 is the best result among ten runs at 20×5 size, in which the black rectangle denotes the breakdown. So, we conclude that QGBSA is very efficient for stochastic FSSP with random breakdown, especially for large-scale with many jobs and machines, which can obtain optimal solution with small iterative generations and population size.

Table 3 is computational results for different cases of TA 20×5 benchmark problem, which are achieved by QGBSA to compute. We found case 1 takes less time than case 3, case 2 takes less time than case 4, which means that the rest part of job being interrupted by breakdown need less processing time than the job to be reprocessed. Under the same working mode, breakdown happened at the life-span time needs more processing time than breakdown happened at any time.

Table 2. Statistical results of QGBSA and GA

Ps	Sampl Times	PS GN	The best	Min/average/max QGBSA	Min/average/max GA
20,20	**20**	**200, 50**	**2297**	2913.32/**2639.52**/2965.49	2910.12/2954.3/2993.35
20, 5	**20**	**200, 50**	**1278**	1499.4/**1523.59**/1572.5	1523.6/1556.45/1602.3
20,10	**20**	**200, 50**	**1582**	1975.3/**2010..96**/2049.08	2001.73/2034.46/2085.94
50,5	**20**	**200, 50**	**2724**	2942.3/**2958.79**/2988.4	2935.3/2962.4/2998.54
50,10	**20**	**200, 50**	**3025**	3568.3/**3634.55**/3685.86	3583.4/3678.36/3720.24

Remark: Ps is the problem size. PS and GN indicate the size of population and the generation of termination respectively. The best means the best solution of the determinate FSSP.

Table 3. Computational results under different breakdown cases

PS GN	Min/average/max Case1	Min/average/max Case3	Min/average/max Case2	Min/average/max Case4
200 50	1484/**1512.7**/1583	1582.6/1573.1/1642.2	1511/**1606.7**/1685.2	1728/1731.3/1915.7
	1504/**1535**/1557.3	1594/1597/1655.1	1508/**1592.8**/1672.7	1668/1709.2/1860.4
	1417/**1495.2**/1543	1590/1592.3/1679.9	1532/**1590.5**/1705.5	1555/1657.3/1796.3
	1503/**1518.3**/1540	1545/1590.8/1700.3	1472/**1585.1**/1714.1	1670/1671.6/1781.5
	1514/**1544.8**/1636	1513/1549.8/1613	1500/**1594.4**/1658.3	1603/1671.6/1791.9
	1498/**1517**/1556.6	1513/1549.8/1713	1489/**1562.8**/1640.5	1563/1661.8/1782.8
	1520/**1550.2**/1578	1533/1612.1/1896.6	1489/**1562.8**/1640.5	1558/1649.6/1780.1
	1524/**1538**/1601.3	1562/1589.2/1764.6	1471/**1566.2**/1698	1527/1649.7/1764.5
	1486/**1516.8**/1601	1569/1579/1599.8	1511/**1590.8**/1697.1	1606/1665.3/1770.6
	1490/**1507.8**/1529	1571/1605.7/1628.8	1509/**1589.5**/1669.2	1566/1681.1/1811.9
AV	1494/**1523.6**/1573	1557.26/1583.8/1689	1499.2/**1444.16**/1674	1604.4/1674/1805.7

Remark: AV is the average value of ten times.

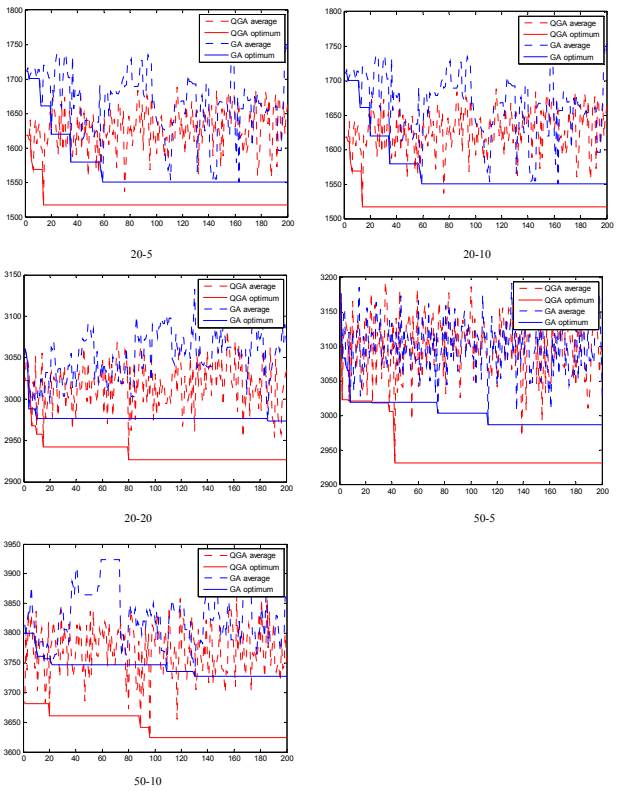

Fig. 3. Convergence curves of different size

6. CONCLUSION

A QGBSA-based Algorithm for stochastic flow shop scheduling with random breakdown problem is proposed in this paper. We consider four different working cases of breakdown. Through simulation on Taillard benchmark problem, compared with GA, the proposed algorithm can achieve a solution of better quality and good robustness on uncertainty for stochastic flow shop scheduling problem. Experiment results show the feasibility and effectiveness of this algorithm.

ACKNOWLEDGEMENT

This work is supported by National Natural Science Found of China under Grant No. 60674075 and No. 60774078 and the key Technologies Program of Shanghai Educational Committee (Grant No. 05ZZ73). It is also supported by Shanghai Leading Academic Discipline Project, Project Number: B504.

REFERENCES

Allaoui, H. and Artiba, A(2004). Integrating simulation and optimization to schedule a hybrid flow shop with maintenance constraints. In: p 431-450, *Computers and Industrial Engineering*.

ALLAHVERDI A, MITTENTHAL J (1995). Scheduling on a two-machine flowshop subject to random breakdowns with a makespan objective function [J]. In: **81 (2)**: 367-387, *European Journal of Operation Research*.

ALCAIDE D and RODRIGUEZ. Other (2002). An approach to solve the minimum expected makespan flow-shop problem subject to breakdowns [J]. In: **140 (2)**: 384-398, *European Journal of Operation Research.*

Bin-Bin Li and Ling Wang (2007). A Hybrid Quantum-Inspired Genetic Algorithm for Multi-objective Scheduling. In: **37 (3)** : 576-591, *IEEE Transactions on Systems, Man, and Cybernetics*.

Li, Su-Fen, Zhu, Yun-Long (2005). Flow shop scheduling with stochastic processing times and machine breakdowns. In: **11 (10)**: 1425-1429, *Jisuanji Jicheng Zhizao Xitong/Computer Integrated Manufacturing Systems*.

M. Gourgand, N. Grangeon, others (2000), A review of the static stochastic flow shop scheduling problem. In: **9(2)** 183-214, *Journal of Decision Systems.*

Xingsheng Gu (2000). Scheduling under uncertainty. In: *Journal of East China university of Science and Technology (Natural Science Edition)*

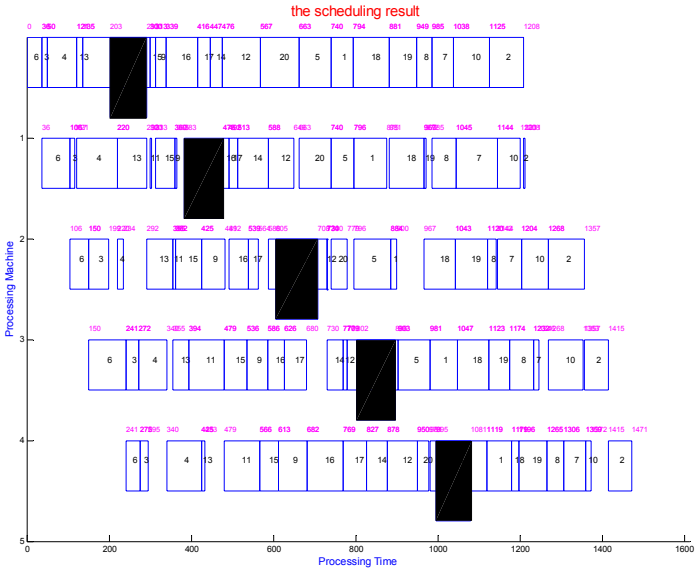

Fig. 4. The gantt chart of 20×5