Multi-objective Optimization of Fuzzy Parallel Machines Scheduling Problem Using Nondominated Genetic Algorithms

Xie Yuan
Shanghai Dian Ji University/Automation Department, Shanghai, China
Email: xiey@sdju.edu.cn

Abstract—**A kind of unrelated parallel machines scheduling problem is discussed. The memberships of fuzzy due dates denote the grades of satisfaction with respect to completion times with jobs. Objectives of scheduling are to maximize the minimum grade of satisfaction while makespan is minimized in the meantime. Two kind of genetic algorithms are employed to search optimal solution set of the problem. Both Niche Pareto Genetic Algorithm (NPGA) and Nondominated Sorting Genetic Algorithm (NSGA-II) can find the Pareto optimal solutions. Numerical simulation illustrates that NSGA-II has better results than NPGA.**

Index Terms—**parallel machines scheduling, fuzzy due-date, NPGA, NSGA-II**

I. Introduction

Scheduling problems have been researched for more than fifty years since first scheduling investigations undertaken by Jackson in 1955[1]. Parallel machines scheduling problem is a kind of important multi-machine scheduling problem. It means every machine has same work function and every job can be processed by any machine. Most of non-preemptive parallel scheduling problems are NP problems and can not be solved with determinative algorithm in polynomial time; even there are only two machines. Genetic algorithm has strong searching ability and robustness. So lots of researchers analyze parallel scheduling problem with genetic algorithm [2, 3, 4, 5].

In the conventional scheduling problem, the parameters such as job processing times, ready times, due-dates have been assumed to be deterministic. In real world, input data may be uncertain or imprecise. Ishii et al. [6] introduced the concept of fuzzy due dates to scheduling problems, fuzzy due dates scheduling problems have been investigated by many researchers [7]. Considering that input parameters of parallel scheduling problems are uncertain in real situation, we fuzzilize the due dates of jobs. For one feasible scheduling sequence, memberships of fuzzy due dates are used to denote the grades of satisfaction about due dates from the point of view of scheduling decision-makers. The objective of scheduling is not only to minimize the maximum completion times of jobs, but also to maximize the minimum grades of satisfaction of jobs' completion times. According to analyzing the bicriteria scheduling problem, Pareto optimal solution is induced into describe the problem. First we use Niche Pareto Genetic Algorithm (NPGA) [8] to find the Pareto optimal solutions of the problem. Then we improve the searching method. Nondominated Sorting Genetic Algorithm (NSGA-II) [9] is employed and shows better result.

The outline of this paper is organized as follows. Firstly, we briefly describe the parallel problem with fuzzy due dates. Then, we introduce the conception of nondominated scheduling. After that, we give the solution of Niche Pareto Genetic Algorithm. We also give another GA based solution—Nondominated Sorting Genetic Algorithm. In order to comparing the two GA based algorithms, some numerical examples and computational results are provided to show the effectiveness of the proposed algorithms. Finally, we conclude this paper with a summary.

II. Description of Problem

Give m machines M_1, M_2, …, M_m and n jobs J_1, J_2, …, J_n. Every machine can process one job at the same time, and arbitrary job can be process on one machine simultaneity. All machines have same function. Then each job can be process on any machine. So we call it parallel machines. The processing time of job J_i is p_{ij} on machine M_j. The m machines are identical machines if there are $p_{ij} = p_j$ for all machine M_j. If the processing time $p_{ij} = p_j/s_j$ on machine M_j for job J_i, which s_j is the speed of machine, machines are uniform machines. If $p_{ij} = p_j/s_{ij}$, which s_{ij} is the speed of job J_i on machine M_j, machines are unrelated machines.

In many practical scheduling problems, due dates of jobs are uncertain or imprecise, and this kind of uncertainty or imprecision cannot be described by probability theory. For these situations where characteristics and constraints are neither deterministic nor probabilistic, the problems may often be modeled with fuzzy sets [10,11].

In scheduling problems, every job has due date d_i. Jobs must be completed before due dates in deterministic scheduling problems. But due dates of jobs are not crisp numbers in lots of practical scheduling. A certain delay is tolerable up to a late date d_i+e_i beyond which the order will be canceled, because the customer will resort to other

doi:10.4304/jsw.6.10.2036-2042

suppliers. As completion time of job J_i passes between due date d_i and late date d_i+e_i, the customer satisfaction decreases until it vanishes at the latter. The greater the delay, the lower the satisfaction. For the interval [d_i, d_i+e_i] in real number set, it is acceptable if job J_i can be finished in its interval. But the grades of satisfaction of scheduling decision-makers about completion time C_i of job J_i are different. We use fuzzy set $\tilde{d}_i$ to describe the uncertain due date. Fuzzy set $\tilde{d}_i$ denotes [d_i, d_i+e_i], its membership $f_i(x)$ is:

$$f_i(x) = \begin{cases} 1 & x \le d_i \\ (d_i + e_i - x)/e_i & d_i < x < d_i + e_i \\ 0 & d_i + e_i \le x \end{cases}. \quad (1)$$

Membership function denotes the grade of satisfaction of completion time C_i(it is x in formula (1)) about due date. It is given by scheduling decision-makers according their scheduling experience or preference. If completion time is not beyond low bound d_i, the grade of satisfaction is 1, which is biggest. The grade of satisfaction declines with increasing of C_i, If C_i is beyond d_i. Until completion time equals or is bigger than upper bound of fuzzy due date d_i+e_i, the grade of satisfaction declines to zero. This membership is shown in Figure 1.

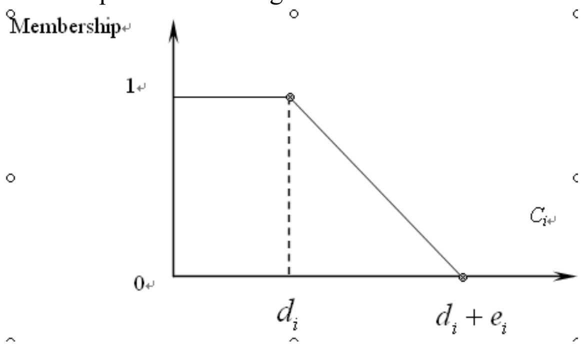

Figure 1. Membership of fuzzy due date for job

For above parallel scheduling problem, we consider two objective functions as following. One is maximum completion time of jobs $C_{\max}$ which is makespan of jobs:

$$C_{\max}=\max\{C_i|i=1,2,\ldots,n\}. \quad (2)$$

Another is minimum grade of satisfaction $f_{\min}(C_i)$ of job completion time C_i:

$$f_{\min}(C_i)=\min\{f_i(C_i)|i=1,2,\ldots,n\}. \quad (3)$$

The objective of scheduling is to find optimal job sequence, which minimizes $C_{\max}$ and maximize $f_{\min}(C_i)$ simultaneity. So the bicriteria scheduling problem can be denoted as $R \mid \tilde{d}_i \mid (C_{\max}, f_{\min})$.

III. No-dominated Scheduling

For above bicriteria scheduling problem $R \mid \tilde{d}_i \mid (C_{\max}, f_{\min})$, the objective is to minimize the maximum job's completion time $C_{\max}$ and maximize the minimum grade of satisfaction $f_{\min}(C_i)$ in the same time. For some feasible schedule π, let schedule vector v^π consisting two elements, $C^\pi_{\max}$ and $f^\pi_{\min}$ in some feasible schedule π. We denote the solution vector as $S^\pi = (C^\pi_{\max}, f^\pi_{\min})$. Given two feasible solutions π_1 and π_2, their vectors are $S^{\pi_1} = (C^{\pi_1}_{\max}, f^{\pi_1}_{\min})$ and $S^{\pi_2} = (C^{\pi_2}_{\max}, f^{\pi_2}_{\min})$, respectively. We say that vector S^{π_1} dominates vector S^{π_2} if

$$C^{\pi_1}_{\max} \le C^{\pi_2}_{\max},\ f^{\pi_1}_{\min} \ge f^{\pi_2}_{\min} \text{ and } S^{\pi_1} \ne S^{\pi_2}. \quad (4)$$

and at least one of these two inequalities is strict. If π_1 and π_2 are such that

$$C^{\pi_1}_{\max} = C^{\pi_2}_{\max},\ f^{\pi_1}_{\min} = f^{\pi_2}_{\min}. \quad (5)$$

then we call them equivalent. Consequently, every equivalence class of schedules consists of all schedules with identical value of $C^\pi_{\max}$ and $f^\pi_{\min}$. During the process of maximize the both criteria $C^\pi_{\max}$ and $f^\pi_{\min}$, maybe there exit many nondominated schedules which have same values. We just need to find at least one feasible schedule corresponding form each equivalence class of nondominated feasible schedules.

For one feasible schedule π, we say it is the nondominated schedule of problem if its vector S^π is not dominated by any other vectors. So the feasible schedule π is the one of optimal schedules whom we are searching for. S^π is called Pareto optimal vector. For scheduling problem $R \mid \tilde{d}_i \mid (C_{\max}, f_{\min})$, it has lots of nondominated schedules who are not dominated by any other feasible schedules. The vectors of those nondominated schedules constitute Pareto optimal frontier of solutions vector space.

IV. The Solution Based on NPGA

Niche Pareto Genetic Algorithm (NPGA). In order to find optimal schedules of problem $R \mid \tilde{d}_i \mid (C_{\max}, f_{\min})$, the corresponding Pareto optimal vectors should be gotten. When all Pareto optimal vectors are given, a Pareto optimal frontier can be constructed in two dimensions space. The different points in the frontier denote corresponding optimal schedules. But different optimal schedules have same Pareto optimal vectors in some case. Horn and Nafpliotis [8] pointed out that Niche Pareto Genetic Algorithm (NPGA) can find the set of Pareto optimal vectors. NPGA algorithm also can keep the diversity of Pareto optimal vectors of every generation and avoid populations to converge into several optimal vectors in population evolution process. The solving process is given as following according to the problem in our research.

Encoding Method. Given parallel machines scheduling problem with n jobs and m machines, n+m-1 genes denote one feasible schedule. Every chromosome is divided into m segments which are separated by m-1 machines M_i(i = 1, 2, . . . , m−1). Here M_i means i-th machine. For example, there are six jobs and three machines. Chromosome "35$M_1$1$M_2$462" denotes that job J_3 is processed on machine M_1 firstly and then J_5 is processed on same machine secondly. In the same time, job J_1 is processed on machine M_2. The left job J_4, J_6 and J_2 are processed on machine M_3 in sequence of $J_4 \rightarrow J_6 \rightarrow J_2$. So M_3 doesn't show in chromosome. If there are no jobs between machines M_i and M_j, it means that M_j is idle. Here some schedules are created randomly. The initial population is created by encoding these schedules in order to get next generation population.

Fitness function. We choose objectives $C_{\max}$ and $f_{\min}(C_i)$ as fitness functions for the bicriteria parallel machines scheduling problem $R \mid \tilde{d}_i \mid (C_{\max}, f_{\min})$.

Selection Operator. Selection operator is critical part in NPGA algorithm. In order to keep the diversity of Pareto optimal schedules in every generation of population and avoid that individual converges to one Pareto optimal schedule; NPGA uses unique tournament selection method. The selection method of NPGA algorithm is following. Firstly, two individuals are chosen randomly from current population as candidates. t_{dom} individuals also are chosen from current population as comparing set P at random. The fitness of each candidate is compared with all individuals of comparing set P. If one candidate is dominated by P and another candidate is not dominated by P, the late one is chosen to reproduction. The sharing to choose the winner if neither or both is dominated by P.

Sharing is a kind of fitness sharing function which is used to maintain individuals to distribute around Pareto optimal frontier. Sharing function $sh[d]$ is the decreasing function about $d[i, j]$, such that $sh[0] = 1$ and $sh[d \geq \sigma_{share}] = 0$. Here $d[i, j]$ is the distance between i and j. Generally speaking, sharing function is denoted by triangle function:

$$sh[d] = \begin{cases} 1 - d / \sigma_{share} & d \leq \sigma_{share} \\ 0 & d > \sigma_{share} \end{cases}. \quad (6)$$

Here σ_{share} is niche radius, given by user. We get the niche count m_i when we add all sharing functions of individual i with other individuals in population:

$$m_i = \sum_{j \in Pop} sh[d[i, j]]. \quad (7)$$

Here m_i means the crowded degree around individual i. In order to avoid population converge into certain individual, sharing function calls for the degradation of an individual's fitness by simply dividing the objective fitness by the niche count m_i. Then we get a new fitness f_i' as following:

$$f_i' = f_i / m_i. \quad (8)$$

Then the fitness of individuals within one niche radius become to decrease and is lower than the fitness of individuals within other niche radius when the quantity of individuals in this niche radius is beyond certain number. So individuals with smaller fitness have less chance to keep in next generation. This guarantees the diversity of final generation of population. So in selection process of NPGA algorithm, the candidate with smaller niche count will be chosen if neither or both of two candidates are dominated by P.

Then choose progress can be described as following. Given S is current population, P is comparing set and consist of t_{dom} individuals. P_i is the ith individual.

Step one: Randomly choose two individuals as candidates from population S. Randomly choose t_{dom} individuals as comparing set P.

Step two: Let $i \leftarrow 1$, comparing two candidates with ith individual P_i. If P_i is better than some candidate, mark the candidate as dominated one. If both of candidates are marked as dominated individuals, go to step four.

Step three: Let $i = i+1$. If $i \leq t_{dom}$, go to step two. Else go to next step.

Step four: If only one candidate is marked as dominated individual, remain another one which is not dominated by comparing set P. If both of individuals are dominated by comparing set P or neither one is dominated by comparing set P, use sharing mechanism.

Crossover Operator. Crossover operator applies partial map crossover (MPX). MPX can be viewed as a variation of two-cut-point crossover. MPX deals with illegal phenomenon according to a special repairing process. The crossover process of MPX is following. Randomly choose two individuals as parents in this generation. Then choose two genes of every parent as cut-point. The substring between two cut-point of every parent directly remains into the same position of corresponding child. The other genes of children can be gotten from another parent according certain mapping relationship.

Mutation Operator. Our research uses following mutation mechanism. Select one gene in every individual at random and divide the individual into two parts. Then swap these two parts. For example, pick the third gene "3" in individual (123456) as swap point. Swap the two parts and get new individual (345612).

Stop Condition. The algorithm stop when given stop condition is reached. Otherwise the algorithm keeps searching the Pareto optimal solutions. Here we use maximum times of propagation as stop condition.

Now the NPGA algorithm for finding the optimal schedules can be described as follows:

Step one: The initial population P_i including N individuals is generated randomly. P_i is current population. Evaluate the fitness of every individual of P_i. We can get the current best individual. Let the best individual be overall best one.

Step two: Randomly choose two individuals from current population as candidate individual. The randomly choose t_{dom} individuals to compose comparing sets P. Using selection operator to generate next generation individuals.

Step three: Repeat step 2 till new population is generated which is consisted of N individuals. The new population replaces the current population.

Step four. Randomly choose two individuals to get new individuals with crossover operator and mutation operator.

Step five: Repeat step 4 till get N new individuals. These N new individuals consist of population P_{i+1}.

Step six: Evaluate the fitness of every individual of P_{i+1}. We can get the best individual and worst individual. Replace the worst individual of P_{i+1} with overall best individual. If the best individual of P_{i+1} is better than overall best individual, replace the overall best individual with the best individual of P_{i+1}.

Step seven: If stop condition is not satisfied, Let P_{i+1} be current population P_i. Else, go to end.

Figure 2 gives the procedure of NPGA algorithm.

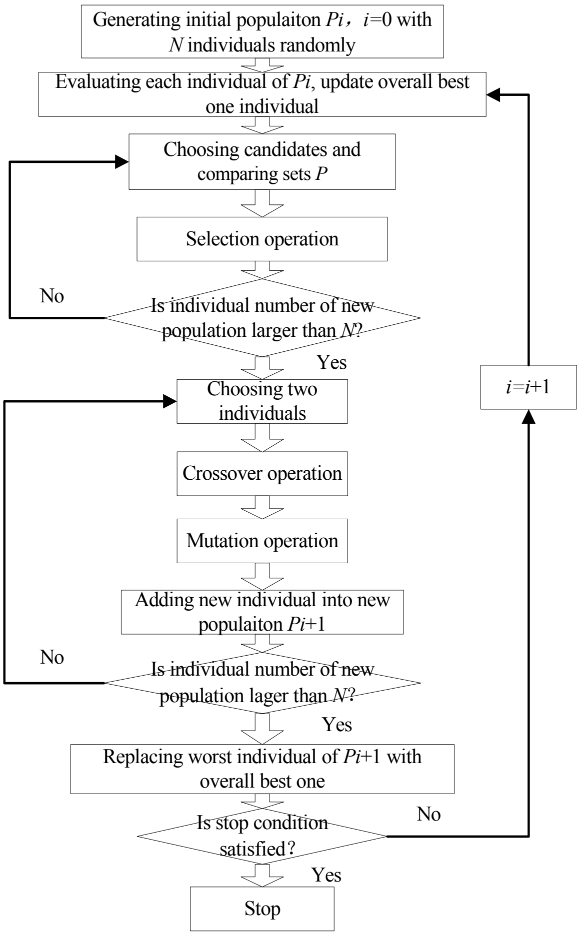

Figure 2. Membership of fuzzy due date for job

V. The Solution Based on NSGA-II

Nondominated Sorting Genetic Algorithm (NSGA-II). NSGA-II algorithm is the modified method of Nondominated Sorting Genetic Algorithm (NSGA) by Deb et al [8]. NSGA algorithm, which is given by Srinivas and Deb, has following shortages:

--First, high computing complexity of NSGA.

--Second, lack of elitism.

--Third, need for specifying the share parameter σ_{share}.

NSGA-II algorithm improves the NSGA algorithm and solves above three shortages successfully. The details of NSGA-II algorithm are following.

Encoding Method. Given parallel machines scheduling problem with n jobs and m machines, n+m-1 genes denote one feasible schedule. Every chromosome is divided into m segments which are separated by m-1 machines M_i(i = 1, 2, . . . , m−1). Here M_i means i-th machine. For example, there are six jobs and three machines. Chromosome "35$M_1$1$M_2$462" denotes that job J_3 is processed on machine M_1 firstly and then J_5 is processed on same machine secondly. In the same time, job J_1 is processed on machine M_2. The left job J_4, J_6 and J_2 are processed on machine M_3 in sequence of J_4→J_6→J_2. So M_3 doesn't show in chromosome. If there are no jobs between machines M_i and M_j, it means that M_j is idle. Here some schedules are created randomly. The initial population is created by encoding these schedules in order to get next generation population.

Fitness function. We choose objectives $C_{\max}$ and $f_{\min}(C_i)$ as fitness functions for the bicriteria parallel machines scheduling problem $R \mid \tilde{d}_i \mid (C_{\max}, f_{\min})$.

Nondominated Schedules. All individuals of current population are sorted according to corresponding fitness functions. Firstly, every Pareto optimal individual is selected in current population. The rank number of these Pareto optimal individuals is one. Then delete individuals whose rank number are one and search Pareto optimal individuals in left population whose rank number are assigned as two. Delete individuals with rank number two and continue to search Pareto optimal individuals in left population. Current Pareto optimal individuals are labeled with rank number three. Repeat above process until every individual has it own rank number. That means individuals have been sorted by its fitness function.

Selection Operator. For NSGA-II algorithm, selection process is very important. In order to keep the diversity of Pareto optimal schedules in every generation of population and avoid that individual converges to one Pareto optimal schedule; NSGA-II algorithm uses unique crowded-comparison approach instead of sharing function of NSGA. The selection strategy of NSGA-II is showed as following. First, NSGA-II algorithm defines density estimation. In order to get density estimation surrounding one individual in the population, NSGA-II algorithm calculates the average distance of two most closed individuals on either side of this individual along each objective. In every generation of population, NSGA-II need choose appropriate individuals from current Pareto optimal individuals into next generation. It means that it depends on density estimation of current Pareto optimal individuals which individuals can be remained in next generation. The total density estimation of individual can be gotten by accumulation all density estimation of its different objective functions. Specifically, current Pareto optimal individuals are ranked from smallest one to biggest one according to the value of this objective function for certain objective function. For biggest and smallest individuals on the end, the density estimation of their objective function is assigned an infinite value. For the other individuals between biggest one and smallest one, their density estimation is absolute normalized difference of the objective function values of two adjacent individuals. Then the total density estimation of one individual can be figured out by accumulation the density estimation of all different functions of this individual.

NSGA-II algorithm is not like Niche Pareto Genetic Algorithm (NPGA) [8] and doesn't require estimating the value of sharing parameter σ_{share} . The latter two algorithms use sharing function to select individuals which can be remained in next generation during selection process. But the number of optimal individuals should be known in advanced if we estimate the value of sharing parameter. For some situation that the distribution of optimal individual can not be known in advanced, it becomes difficult to select sharing parameter. Otherwise the value of sharing parameter has great influence to final result.

Crossover Operator. Crossover operator applies partial map crossover (MPX). MPX can be viewed as a variation of two-cut-point crossover. MPX deals with illegal phenomenon according to a special repairing process. The crossover process of MPX is following. Randomly choose two individuals as parents in this generation. Then choose two genes of every parent as cut-point. The substring between two cut-point of every parent directly remains into the same position of corresponding child. The other genes of children can be gotten from another parent according certain mapping relationship.

Mutation Operator. Our research uses following mutation mechanism. Select one gene in every individual at random and divide the individual into two parts. Then swap these two parts. For example, pick the third gene "3" in individual (123456) as swap point. Swap the two parts and get new individual (345612).

Elitist selection strategy. Deb et al point out that elitist selection strategy can dramatically speed up converging speed and performance and keep all current Pareto optimal individuals remain in next generation of population. It can enhance performance of algorithm and speed up that population converges into Pareto optimal solutions.

Stop Condition. The algorithm stop when given stop condition is reached. Otherwise the algorithm keeps searching the Pareto optimal solutions. Here we use maximum times of propagation as stop condition.

Then NSGA-II algorithm can be described in figure 3 as following.

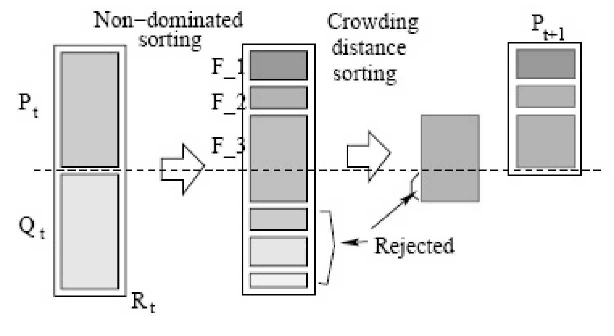

Figure 3. The solution of NSGA-II

Step one: Create initial population P_t, t=1 with N individuals at random, which P_t is current population.

Step two: Randomly choose two individual as parents and use crossover and mutation operator to get two new individual. Repeat above selection, crossover and mutation process till N new individuals are created. These N new individuals compose population Q_t.

Step three: Combine the population P_t and Q_t to get bigger population R_t. Sort all individuals in R_t according density estimation and get biggest rank number F_1.

Step four: Choose all individuals whose rank number is one to remain in next generation of population P_{t+1}. If there are no N individuals, continue to choose the individuals with next rank number. When it goes to rank number F_n, go to step six if the size of population P_{t+1} exactly reaches N. And go to step five if individuals are more than N.

step five: Calculate the density estimation of all individual with rank number F_n. Put right individuals into population P_{t+1} according to descending order of density estimation of individuals till the number of individuals in population P_{t+1} is exactly N.

Step six: Set P_{t+1} as current population P_t and go to step two if stop condition doesn't been reached.

VI. Simulation Example

In order to validate the effectiveness of NSGA-II algorithm, we use Visual C++6.0 to realize the algorithm

TABLE I.
Results of four problems with NAGA-II and NPGA

n×m	NPGA				NSGA-II			
	A	T(s)	C_{max}	f_{min}	A	T(s)	C_{max}	f_{min}
30×3	4	11	147.00	0.415	2	12	147.00	0.415
			148.00	0.662			150.00	0.701
			149.00	0.670				
			153.00	0.729				
40×4	2	12	118.00	0.852	2	12	120.00	0.855
			122.00	0.860			125.00	0.833
50×5	3	13	100.00	0.833	3	13	99.00	0.814
			106.00	0.849			105.00	0.854
			110.00	0.861			109.00	0.858
60×6	2	14	192.00	0.370	5	14	181.00	0.271
			195.00	0.648			185.00	0.322
							190.00	0.357
							194.00	0.579
							197.00	0.684

n×m means the size of problem. A is quantity of optimal individuals. T is running time of computer for the responding problem, whose unit is second.

and make several example to simulate it on computer with PIII CPU. The size of problems n×m (n jobs and m machines) is 30×3, 40×4, 50×5 and 60×6, respectively. The processing times of jobs are integer numbers of interval [0, 50] gotten at random. Randomly create one scheduling sequence S and calculate the completion times $C_i(S)$ of each job J_i. The low bound of fuzzy due date $\tilde{d}_i$ of job J_i is denoted by $d_i= C_i(S)-x$, which x is the random number in the interval [1, 100]. The span e_i of fuzzy due date $\tilde{d}_i$ also is the random number of close interval [1, 100]. Above-mentioned four problems are NP-hard problems even in the case of single objective function. Here we also use Niched Pareto Genetic Algorithm (NPGA) of Horn and Nafpliotis to solve these four instances. We compare the resolution and running time of NPGA and NSGA-II. The two algorithms consider maximum completion time and fuzzy due date of jobs simultaneity when the size of population is 100 individuals. We set that crossover probability is 0.8 and mutation probability is 0.1. In NPGA algorithm, the quantity of individuals in comparing set is ten. And the shape of niche of sharing function is 0.5. Then results are showed as following table 1.

VII. SUMMARY

Our research discusses a kind of parallel machine scheduling problem with fuzzy due dates. Fuzzy due dates mean those due dates of jobs are not crisp numbers and can vary in certain interval. It is acceptable if the completion times of jobs are in this interval. But for scheduling decision-makers, the grades of satisfaction of jobs completion times are different. So it makes jobs scheduling more flexible and realistic. Here the purpose of scheduling is to minimize the maximum completion times of jobs and maximize the minimum grades of satisfaction of fuzzy due dates. For the bicriteria scheduling problem, the concept of nondominated schedules is introduced. The solution vector $(C_{\max}, f_{\min})$ of nondominated schedule doesn't dominated by the solution vector of any other schedules. So these nondominated solution vectors are Pareto optimal solutions. All Pareto optimal solutions construct the set of Pareto optimal solutions.

NPGA algorithm can solve scheduling problem in our research. But NPGA algorithm requires that scheduling decision-makers give shape of niche or estimate shape of niche. Scheduling decision-makers also should give individual quantity of comparing set in tournament selection. These two parameters have some influence to NPGA algorithm. But there is not deterministic mathematical method or formula to compute these two parameters. It depends on experience to determine them. It affects the usage of NPGA algorithm in practical scheduling application. So we apply NSGA-II algorithm to search the Pareto optimal solutions of parallel machines scheduling problem with fuzzy due dates. NSGA-II algorithm uses crowed-comparison approach to keep appropriate individuals into next generation. It is different from sharing function of NPGA. Crowed-comparison approach doesn't require estimating the shape of niche. It selects individuals by computing ranks of nondominated sets of individuals and crowed distances of individuals. This method makes selection operator more clear and convenient. Moreover, crowed-comparison approach needn't choose comparing set and avoid estimate size of comparing set. All of above advantages make NSGA-II algorithm is easier to use and less computation than NPGA. The original version of NSGA-II algorithm given by Deb requires storing dominated individual set of every individual and it needs $O(n^2)$ space complexity. Our research modified original version and give a kind of computing method with smaller space complexity. The new method makes space complexity reduce to $O(n)$.

ACKNOWLEDGMENT

This work was partially supported by Outstanding Young Teachers Project of Shanghai Municipal Education Commission (sdj09001), Special Project of Science and Technology Commission of Minhang District, Shanghai (2010MH075), Project of Science and Technology Commission of Shanghai (10JC1405800, 08DZ1200505), sub-project of Shanghai Economic Commission (09A118), key discipline of Shanghai Municipal Education Commission (J51901).

REFERENCES

[1] J.R. Jackson, "Scheduling a production line to minimize maximum tardiness," *Management Science Research Project*. UCLA. 1955.

[2] S. Samur, and S. Bulkan, "Three Genetic Algorithm Approaches to Parallel Machine Scheduling and Comparison with a Heuristic," *Information Technology: New Generations (ITNG), 2010 Seventh International Conference on*. pp. 270-274, October 2010.

[3] G. Abaza, I. Badr, P. Goehner, and S. Jeschke, "Extending an agent-based FMS scheduling approach with parallel genetic algorithms," *IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society*. pp. 2689-2694, November 2010.

[4] A. Mihaila, and C. Mihaila, "Uniform Parallel Machines Scheduling Using a Genetic Algorithm," *Intelligent Systems Design and Applications, 2008. ISDA '08. Eighth International Conference on*, vol. 2, pp. 401-406, November 2008.

[5] Xie Yuan, Ou linlin, "Bicriteria Parallel Machines Scheduling Problem With Fuzzy Due Dates Based On NPGA," *In Proceedings of the Eighth International Conference on Machine Learning and Cybernetics*, pp.12-15, Baoding, July 2009.

[6] H. Ishii, M.Tada, and T. Masuda, "Two Scheduling Problems with Fuzzy Due-dates," *Fuzzy Sets and Systems*. Vol. 46, pp. 339-347, March 1992.

[7] I. Gonzalez-Rodriguez, J. Puente, C. R. Vela, and R. Varela, "Semantics of Schedules for the Fuzzy Job-Shop Problem," *European Journal Operational Research*, Vol. 79, pp. 1-12, December 1994.

[8] Jeffrey Horn, Nicholas Nafpliotis, and David E. Goldberg, "A niched pareto genetic algorithm for multiobjective optimization," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on*. vol 38, pp. 655-666, May 1994.

[9] Kalyanmoy Deb, Amrit Pratap, Sameer Agarwal, and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II," *IEEE Trans. Evolutionalry Computation*. Vol. 6, pp. 182-197, April 2002.

[10] F. Herrea, and J. L. Verdegay, "Fuzzy sets and operations research: perspectives," *Fuzzy sets and Systems*, Vol. 90, pp. 207-218, September 1997.

[11] D. Dubois, H. Fargier amd P. Fortemps, "Fuzzy scheduling: modelling flexible constraints vs. coping with incomplete knowledge," *European Journal of Operational Research*. Vol. 147, pp. 231-352, June 2003.

Xie Yuan was born in Jiangxi Province, China in Apr. 24, 1978. Mr. Xie received his B.S. degree in precision instrument department of Heifei University of Technology, Heifei, China, in 1999. Mr. Xie holds a M.S. degree in mechanical engineering from Heifei University of Technology. Mr. Xie received his Ph.D. degree from automation department of Shanghai Jiao Tong University, Shanghai, China, in 2006.

He was an engineer of system integration and management department of Shanghai Maglev Transportation Development from 2006 to 2007, and was R&D engineer of automation Department of SANY Electricity Co. Ltd. Currently, he is a Lecturer of automation department of Shanghai Dian Ji University, Shanghai, China. His research interests are wind power technology, fault diagnoses and intelligent algorithm.