Journal of Industrial Engineering International *Islamic Azad University, South Tehran Branch*
August 2006, Vol. 2, No. 3, 43-53

An algorithm for multi-objective job shop scheduling problem

Parviz Fattahi*
Ph.D. Candidate, Dep. of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Mohammad Saidi Mehrabad
Associate Professor, Dep. of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Mir B. Aryanezhad
Professor, Dep. of Industrial Engineering, Iran University of Science and Technology, Tehran, Iran

Abstract

Scheduling for job shop is very important in both fields of production management and combinatorial optimization. However, it is quite difficult to achieve an optimal solution to this problem with traditional optimization approaches owing to the high computational complexity. The combination of several optimization criteria induces additional complexity and new problems. In this paper, we propose a Pareto approach to solve multi-objective job shop scheduling. The objective considered is to minimize the overall completion time (makespan) and total weighted tardiness (TWT). An effective simulated annealing algorithm based on proposed approach is presented to solve multi-objective job shop scheduling problems. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the problem solving process. The parameters in the proposed algorithm are determined after conducting a pilot study. Numerical examples are used to evaluate and study the performance of the proposed algorithm.

Keywords: Job shop; Scheduling; Pareto approach; Simulated annealing; Multi-objective

1. Introduction

The problem of scheduling in job shops has been extensively investigated by many researches. The job shop scheduling problem is to determine a schedule of jobs that have pre-specified operation sequences in a multi-machine environment. In the classical job shop scheduling problem (JSP), n jobs are processed to completion on m unrelated machines. For each job, technological constraints specify a complete, distinct routing which is fixed and known in advance. Processing times are fixed and known in advance. Each machine is continuously available from time zero, and operations are processed without preemption.

Since scheduling began to be studied at the beginning of this century, numerous papers have been published. Almost all of them optimize a single objective. Many industries such as aircraft, electronics, semiconductors manufacturing, etc., have tradeoffs in their scheduling problems where multiple objectives need to be considered in order to optimize the overall performance of the system. Optimizing a single objective generally leads to deterioration of another one. For example, increasing the input rate of product into a system generally leads not only to higher throughput, but also to increased work-in-process (WIP).

The most of the contributions reported in the literature dealing with multi-objective scheduling problems have been divided to these categories:

- Review of the multi-criteria scheduling problem. Hoogeveen [8] presented a comprehensive review of the published literature on the multi-criteria scheduling. He presented that the following performance criteria appeared

* *Corresponding author. E-mail: fattahi@basu.ac.ir*

frequently in the literature: maximum completion time or makespan $(C_{\max}(\sigma))$, total weighted completion time $(\sum_{j=1}^{n} w_j c_j(\sigma))$, maximum lateness $(L_{\max}(\sigma) = \max_j L_j(\sigma))$, maximum tardiness $(T_{\max}(\sigma) = \max_j T_j(\sigma))$, maximum cost $(f_{\max}(\sigma) = \max_j f_j(C_j(\sigma)))$, total weighted tardiness $(\sum_{j=1}^{n} w_j T_j(\sigma))$, maximum earliness $(E_{\max}(\sigma) = \max_j E_j(\sigma))$, total weighted earliness $(\sum_{j=1}^{n} w_j E_j(\sigma))$, weighted number of tardy jobs $(\sum_{j=1}^{n} w_j U_j(\sigma))$.

- Pareto approach for multi-objective scheduling. In such multi-objective scheduling problems, it is common to obtain a set of Pareto-optimal or efficient solutions such that at least one such solution is not inferior to any other given solution not contained in the set, and the solutions in the set do not dominate each other. This approach is applied for single machine scheduling [6,10], flow shop scheduling [17] and parallel machines [2].
- Scalar approach for multi-objective scheduling. One common approach in dealing with such situations is to establish a weighted (composite) objective function based on the significance of individual objectives, or equivalently, the criticality of deviating from the optimal value of each individual objective. This approach is applied for single machine scheduling [5,6] and job shop scheduling [19].
- Various objectives in job shop scheduling. The following performance criteria appeared frequently in the single objective job shop scheduling literature: maximum completion time or makespan [3,13,18], various tardiness objectives [1,11], penalty cost [7] and various earliness objectives [15]. Two objectives representing the general performance of a manufacturing system are considered in this study. They are minimizing makespan and minimizing total weighted tardiness (TWT).

All the above problems are either shown to be NP-hard or remain open as far as computational complexity is concerned. Obviously the problems including more than one criterion are more difficult. The multi-objective scheduling is strongly NP-hard and combinatorial. No method is able to generate optimal solutions for the multi-objective case in polynomial time. This limits the quality of design and analysis that can be accomplished in a fixed amount of time. For this reason many studies have focused on developing heuristic procedures for this problem. Effectively, metaheuristics, like simulated annealing (SA), tabu search and genetic algorithms have demonstrated their ability to solve combinatorial problems. So, some authors suggested adapting metaheuristics in order to solve multi-objective combinatorial (MOCO) problems [4]. In particular, Ulungu et al. [16] conceived a multi-objective simulated annealing (MOSA) algorithm for solving combinatorial optimization problems.

In this paper, we consider the problem of developing heuristically efficient (or non-dominated) solutions with the objectives of minimizing the overall completion time (makespan) and total weighted tardiness (TWT) of jobs. A Pareto approach based on simulated annealing algorithm is presented to solve the multi-objective job-shop scheduling problem. The aim is to generate a good set of approximation non-dominated solutions. The procedure is valid for any number of objectives. The paper is organized as follows: the problem description and the multiple objectives job-shop scheduling model is described in Section 2 and the notations are introduced. Section 3 gives a description of the MOSA heuristic and the solution procedure. Section 4 reports some computational results and their analysis; conclusions and further research directions are presented in Section 5.

2. Problem description

2.1. Job shop scheduling problem

The scheduling problem under consideration has m machine and n jobs. Each job consists of a sequence of operation $O_{j,h}, h = 1,\ldots,h_j$, where $O_{j,h}$ and h_j denote that hth operation of job j and the number of operations required for job j, respectively. Unless stated otherwise, index i denotes a machine, index j denotes job, and h denotes operation throughout the paper. The due date d_j of job j is promised to its customer, and a delay penalty is charged if job j is completed after its due date. Let w_j denote the delay penalty per unit time of job j. The problem is thus to determine a sequence of the operation on all machines that minimize some criteria.

Most of the research in job shop scheduling is concerned with the minimization of a single criterion. However, scheduling problems often involve more than one aspect and therefore require multiple criteria analysis. Despite their importance, scant attention has been given to multiple criteria scheduling problems, especially in the case of multiple machines.

Two objectives representing the general performance of a manufacturing system are considered in this study. They are minimizing makespan and minimizing total weighted tardiness (TWT). In the following, we use the classical notations:

C_j	The completion time for job *j*.
d_j	The due date for job *j*.
w_j	A possible weight associated to job *j*.
T_j	The tardiness of job *j*.
TWT	The total weighted tardiness.
$C_{\max}$	The maximum completion time (makespan).

Where

$$T_j = \max(0, C_j - d_j),$$

$$TWT = \sum_j w_j T_j ,$$

$$C_{\max} = \max_j (C_j).$$

2.2. Multi-objective optimization

We consider a general optimization problem with two objectives, where we want to minimize functions $f_1(x)$ and $f_2(x)$ subject to a constraint $x \in S$. We denote the vector of objective functions by:

$$F(x) = (f_1(x), f_2(x))^T .$$

The vector $x = (x_1, x_2, ..., x_n)$ is called a decision vector and $S \subset \Re^n$ is the feasible region. The feasible region is formed by constraint functions. The image of the feasible region $Z = F(S)$ is called the feasible objective region. Vectors belonging to the feasible objective region Z are called objective vectors and they are denoted by $F(x) \in \Re^2$.

We want to minimize simultaneously both objective functions. Generally, it is not possible to find a solution in which both objective functions attain minimum values. This means that the objective functions are conflicting. Besides, the feasible objective region *Z* is only partially ordered. In other words, we cannot compare all the objective vectors mathematically. For example, we cannot distinguish which is a better objective vector, $(1,5)^T$ or $(5,1)^T$. However, we can say that $(1,5)^T$ is better than $(2,5)^T$ or $(1,6)^T$.

This leads us to the concept of Pareto optimality. A decision vector $x^* \in S$ and the corresponding objective vector $F(x^*)$ are Pareto optimal if there does not exist another decision vector $x \in S$ such that $f_i(x) \le f_i(x^*)$ for i=1,2 and $f_i(x) < f_i(x^*)$ for at least one *i* [12]. A set containing all the Pareto optimal solutions of the problem is called the Pareto optimal set or non-dominated solutions set. As an example, in Figure 1, we consider a two objective functions case. The solutions *C, D* and *F* are dominated and *{A, B, E, G}* is the Pareto-optimal set of solutions. The main aim of such an approach is to find all the elements of this set in order to give more choice to the decision-maker [9].

Now the solution we are looking for is a non-dominated solution set. This guarantees that we cannot improve any of the objective function values of the solutions without deteriorating the other objective function value. This, which Pareto-optimal solution is the best, depends usually on a decision maker. So, we present an algorithm that searches the non-dominated solutions set for the multi-objective optimization problem considered.

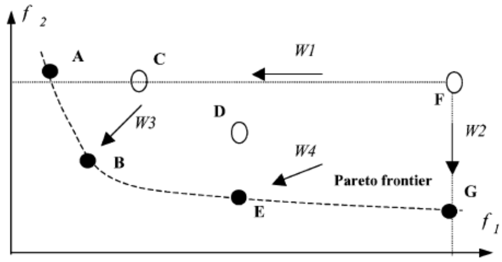

Figure 1. An illustration example of a non-dominated set.

2.3 Mathematical model

Under these assumptions and notations, the problem is to find a schedule that minimizes the overall completion time (makespan) and total weighted tardiness (TWT) given $n, m, O_{j,h}, a_{i,j,h}, d_j$ and $p_{j,h}$.

$$a_{i,j,h} = \begin{cases} 1 & \textit{if } O_{j,h} \textit{ is performed on machine } i \\ 0 & \textit{otherwise} \end{cases}$$

T Total tardiness of schedule.

M A large number.

$$a_{i,j,h,k} = \begin{cases} 1 & \textit{if } O_{j,h} \textit{ is performed on machine } i \textit{ in priority } k \\ 0 & \textit{otherwise} \end{cases}$$

$t_{j,h}$ Start time of the processing of operation $O_{j,h}$.

$Tm_{i,k}$ Start of working time for machine i in priority k.

T_j The tardiness of job j.

A mixed integer program for the multi-objective job shop scheduling problem is then given as:

Min $T = \sum_{j=1}^{n} w_j T_j$

Min $C_{\max}$

Subject to:

$$T_j \geq \max((t_{j,h_j} + p_{j,h_j}) - d_j, 0) \quad j = 1,\ldots,n \quad (1)$$

$$t_{j,h} + p_{j,h} \leq t_{j,h+1} \quad j = 1,\ldots,n, h = 1,\ldots,h_j - 1 \quad (2)$$

$$C_{\max} \geq t_{j,h_j} \quad j = 1,\ldots,n \quad (3)$$

$$Tm_{i,k} + p_{j,h} x_{i,j,h,k} \leq Tm_{i,k+1} \quad i = 1,\ldots,m, \quad j = 1,\ldots,n, \; h = 1,\ldots,h_j - 1, \; k = 1,\ldots,k_i - 1 \quad (4)$$

$$Tm_{i,k} \leq t_{j,h} + (1 - x_{i,j,h,k})M \quad i = 1,\ldots,m, \quad j = 1,\ldots,n, \quad h = 1,\ldots,h_j, \quad k = 1,\ldots,k_i \quad (5)$$

$$Tm_{i,k} + (1 - x_{i,j,h,k})M \geq t_{j,h} \quad i = 1,\ldots,m, \quad j = 1,\ldots,n, \quad h = 1,\ldots,h_j, \quad k = 1,\ldots,k_i \quad (6)$$

$$x_{i,j,h,k} \leq a_{i,j,h} \quad i = 1,\ldots,m, \; j = 1,\ldots,n, \; h = 1,\ldots,h_j, \; k = 1,\ldots,k_i \quad (7)$$

$$\sum_j \sum_h x_{i,j,h,k} = 1 \quad i = 1,\ldots,m, \; k = 1,\ldots,k_i \quad (8)$$

$$\sum_i \sum_k x_{i,j,h,k} = 1 \quad j = 1,\ldots,n, \; h = 1,\ldots,h_j \quad (9)$$

$$t_{j,h} \geq 0 \quad j = 1,\ldots,n, \quad h = 1,\ldots,h_j \quad (10)$$

$$x_{i,j,h,k} \in \{0,1\} \quad i = 1,\ldots,m, \; j = 1,\ldots,n, \; h = 1,\ldots,h_j, \; k = 1,\ldots,k_i. \quad (11)$$

Constraint (1) determines the tardiness of each job. Constraint (2) enforces each job to follow a specified operation sequence. Constraint (3) determines the makespan of schedule. Constraint (4) forces each machine to process one operation at a time. Constraints (5) and (6) force each operation $O_{j,h}$ can be start after its assigned machine is idle and previous operation $O_{j,h-1}$ is completed. Constraint (7) determines the machine for each operation. Constraint (8) assigns the operations to a machine and sequence assigned operations on all machines. Constraints (9) force each operation can be performed only on one machine and one priority. Results of $x_{i,j,h,k}$ yield an assignment each operation on a machine and sequence assigned operations on all machines.

3. Solution procedure

3.1. Simulated annealing

Simulated annealing (SA) is a neighborhood search technique that has produced good results for combinatorial problems. A standard SA procedure begins by generating an initial solution at random. At each stage, the new solution taken from the neighborhood of the current solution is accepted as the new current solution if it has a lower or equal cost; if it has a higher cost, it is accepted with a probability that decreases as the difference in the costs increases and as the temperature of the method decreases. This temperature, which is simply a positive number, is periodically reduced by a temperature scheme, so that it moves gradually from a relatively high value to near zero as the method progresses. Thus, at the start of SA most deteriorating moves are accepted, but at the end only ameliorating ones are likely to be accepted. The method converges to a local optimum as the temperature approaches zero, but because SA has performed many perturbations at higher temperatures which have pushed the search path into new areas, a better local optimum solution should hopefully be reached. In this paper, additional termination criteria are introduced into the proposed SA procedure to reduce the computational effort. The entire SA procedure is terminated either when the temperature T reaches a pre-specified value T_f or the frozen counter f_c is greater than a pre-defined value. In addition, to intensify and diversify the SA searching procedure, when a solution is not ameliorated in a pre-defined number of consecutive temperature stages, a "restarting solution" mechanism is designed to generate a new schedule *S* for the further amelioration of the solution.

3.2 The multi-objective simulated annealing algorithm

The following procedure is used to applying Simulated Annealing metaheuristic for solving the considered problem. In Step 1 an initial solution is generated. SA is used to determine the best sequences of jobs for any machine in Step 2. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the solution process.

Denote *D* is the non dominated solution.

The MOSA algorithm applied in this study is now described as follows:

Step 1. Initialization.

- *1.1.* Obtain an initial solution *S*.
- *1.2.* Compute the objectives of *S* and set $S \rightarrow D$ (inter *S* to *D*).
- *1.3.* Initiate the initial temperature *T*, final temperature T_f, and cooling rate *r*.
- *1.4.* Set the counter markov = 0.

Step 2. While not yet frozen, $(T > T_f)$ or $(n_{bsc} < 50)$, do the following:

- *2.1.* If markov = U_markov, generate a restarting solution *S*, and set markov = 0.
- *2.2.* Perform the following loop *L* times.
  - *2.2.1.* Neighborhood search. Select a neighbor S^c of S.
  - *2.2.2.* Compute $\Delta = R_{S^c} - R_s$. A scalar approach is used to convert the multi-objective problem to a single objective problem.
  - *2.2.3.* If $\Delta \le 0$ set $S = S^c$.
  - *2.2.4.* Compute $\Delta_b = R_s - R_{s^*}$.
  - *2.2.5.* If *S* is non-dominated, set $S \rightarrow D$, update the non-dominated set (*D*) and $n_{bsc} = 0$.
  - *2.2.6.* If $\Delta > 0$, select a random variable $P \sim U(0,1)$. If $e^{-\Delta/T} > P$, set $S = S^c$.
- *2.3.* If the *D* is not updated, set counter markov = markov + 1.
- *2.4.* Set $T = r \times T$.

Step 3. Return the non-dominated solutions.

3.3 The elements of the simulated annealing considered

The method proposed by Saidi and Fattahi [14] is used to represent the solution seed. They name all operations for a job with the same symbol and then interpret them according to the order of occurrence in the sequence for a given solution seed.

Each job appears in the solution exactly h_j times and each repeating number does not indicate a concrete operation of a job but refers to an operation which is context-dependent. The initial solution is obtained with a random operator. An example of the solution seed for a two job - two machine job shop problem, Job 1 has two operations and Job 2 has three operations, is as follows:

[1 2 2 1 2]

Given a sequence *S*, a new sequence S^c is obtained for *S* using a general pairwise interchange: randomly select two positions *i* and *j* in the sequence. A neighborhood of *S*, S^c is obtained by interchanging the jobs in position *i* and *j*.

To make the performance of the proposed SA algorithm more robust, parameter setting is necessary. In this research the parameter setting for the designed SA heuristic is stated in detail. From the results of some preliminary experiments, the initial temperature ranges from 100 to 300, depending on the size of the problem concerned; and the final temperature is determined to be 0.1 for all the cases of the problem. A temperature reduction factor r of 0.99 has been chosen. The upper limit of the frozen counter, that the solution is not improved in a number of consecutive temperature stages, U_markov, is set to be 20 to 50.

As previously noted, an intensification scheme often takes the form of reinforcing attributes of random solutions while a diversification scheme typically consists of driving the search into regions not yet explored.

4. Numerical experiments

This section describes the computational experiments which are used to evaluate the performance of the proposed algorithm in finding good quality schedules. Four problems (MOJ1:4.3.3, MOJ2:6.4.4, MOJ3:10.4.6 and MOJ4:15.4.7 that are shown in Table1s 1, 2, 3 and 4) of multi-objective job shop scheduling based on practical data have been selected to evaluate the performance of the proposed algorithm. These problems are represented by MOJ (multi-objective job shop), n (number of jobs), h (number of operations) and m (number of machines).

These problems are solved by the proposed algorithm to evaluate the performance of it. The algorithm was run on a PC that has a Pentium-IV 1.89 GHz processor, with 512 Mb RAM. All of the problems are solved by the proposed algorithm. Their results are shown in Table 5. Moreover, we presented a mathematical model (Section 2.3) to describe the problem considered and achieve two optimum solutions for each problem. For this reason, we use scalar approach to decrease the complexity of the mathematical model. In this approach, a vector $[u_1, u_2]^T$ is used to transform the multi-objective problem to a single objective problem which described below. To achieve two Pareto solutions for a problem, we use $[1,0]^T$ and $[0,1]^T$ vectors. With any of other vectors $([\alpha, 1-\alpha]^T, 0<\alpha<1)$, may be a Pareto solution is obtained. All of solutions with all of vectors must be cheeked to determine other optimal Pareto solutions. The single objective models are solved with the branch and bound method and their results are shown in Table 6. Solving of the job shop scheduling problem with the branch and bound method is very time consuming and no method is able to generate optimal solutions for the multi-objective case in polynomial time. This limits the quality of design and analysis that can be accomplished in a fixed amount of time. For this reason, many studies have focused on developing heuristic procedures for this problem.

$$F(x) = [f_1(x), f_2(x)] \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{12}$$

A review of the results in Tables 5 and 6 shows that the proposed algorithm is capable to obtain the solution near the optimal solution. Moreover, the proposed algorithm can obtain all of Pareto solution in a small time as shown in Figure 2. Therefore, the proposed algorithm is useful in multi-objective job shop scheduling problems. The non-dominated set for the problem MOJ4 is obtained through the solution process and shown in Figures 3 and 4. These figures show that the non-dominated set will be updated during the solution process and the final non-dominated set will be presented.

5. Conclusion

In this paper, we have dealt with multi-objective job shop scheduling problems and we have proposed a new Pareto-optimally approach to solve them. An effective simulated annealing algorithm based on proposed approach is presented. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the solution process. The objective functions considered are the minimization of the total weighted tardiness (TWT) and Makespan $(C_{\max})$. The obtained results show the efficiency of the proposed algorithm. Although it dose not guarantee the optimality, such an approach provides good quality solutions in a reasonable time limit as future research direction, applying another objective to this problem can be considered.

Table 1. Problem MOJ 1: 4.3.3.

		Operation 1		Operation 2		Operation 3	
Jobs	**Due date**	**Machine number**	**Process time**	**Machine number**	**Process time**	**Machine number**	**Process time**
1	150	1	85	2	55	2	95
2	190	1	26	2	54	3	83
3	140	2	45	3	23	3	68
4	480	2	47	2	58	3	42

Table 2. Problem MOJ 2: 6.4.4.

		Operation 1		Operation 2		Operation 3		Operation 4	
Jobs	**Due date**	**Machine number**	**Process time**	**Machine number**	**Process time**	**Machine number**	**Process time**	**Machine number**	**Process time**
1	450	1	25	2	55	3	45	4	45
2	650	1	26	2	54	3	55	4	23
3	160	2	40	3	25	2	60	3	55
4	350	1	30	2	75	4	60	4	62
5	440	2	45	2	23	3	25	3	68
6	220	1	47	3	25	3	35	4	42

Table 3. Problem MOJ 3: 10.4.6.

		Operation 1		Operation 2		Operation 3		Operation 4	
Jobs	**Due date**	**Machine number**	**Process time**	**Machine number**	**Process time**	**Machine number**	**Process time**	**Machine number**	**Process time**
1	650	1	25	2	55	4	45	5	45
2	450	1	26	2	54	5	55	6	23
3	330	2	40	3	25	5	60	6	55
4	300	1	30	2	75	4	60	5	62
5	1000	2	50	3	45	5	45	6	65
6	250	2	45	3	65	4	25	5	47
7	350	3	50	3	35	5	34	6	26
8	550	2	47	4	20	4	25	5	54
9	250	2	45	2	23	6	25	5	68
10	850	1	47	3	25	5	35	6	42

Table 4. Problem MOJ 4: 15.4.7.

		Operation 1		Operation 2		Operation 3		Operation 4	
Jobs	Due date	Machine number	Process time	Machine number	Process time	Machine number	Process time	Machine number	Process time
1	700	1	25	3	55	4	45	7	45
2	850	1	58	3	54	5	75	7	93
3	300	2	40	2	25	5	60	6	55
4	350	1	30	2	85	4	60	6	92
5	220	2	50	3	45	5	45	7	65
6	400	2	45	2	65	4	25	7	47
7	790	3	90	2	95	5	104	6	126
8	650	2	47	3	20	4	25	6	54
9	750	2	45	4	23	6	25	6	68
10	390	3	55	2	45	4	55	6	55
11	560	3	45	3	65	4	36	7	47
12	470	2	35	3	32	5	45	7	48
13	900	1	62	2	33	5	65	6	35
14	400	2	28	3	25	6	33	6	65
15	230	1	47	3	25	5	35	7	42

Table 5. The final non-dominated sets for the experimental problem.

MOJ 1		MOJ 2		MOJ 3		MOJ 4	
Makespan	TWT	Makespan	TWT	Makespan	TWT	Makespan	TWT
354	233	407	247	596	167	868	1344
408	140	484	33	590	419	883	1285
396	175	487	32	580	572	826	2769
		459	40	583	533	831	2385
		433	78			842	2039
		430	138				
		445	75				
		494	20				
CPU time: 15		45		125		385	

Table 6. Results of the mathematical model and the branch and bound method.

MOJ 1		MOJ 2		MOJ 3		MOJ 4	
Makespan	TWT	Makespan	TWT	Makespan	TWT	Makespan	TWT
354	233	407	247	596	167	Not available	
408	140	494	20	580	572		
CPU time: 650		980		4700		>7200	

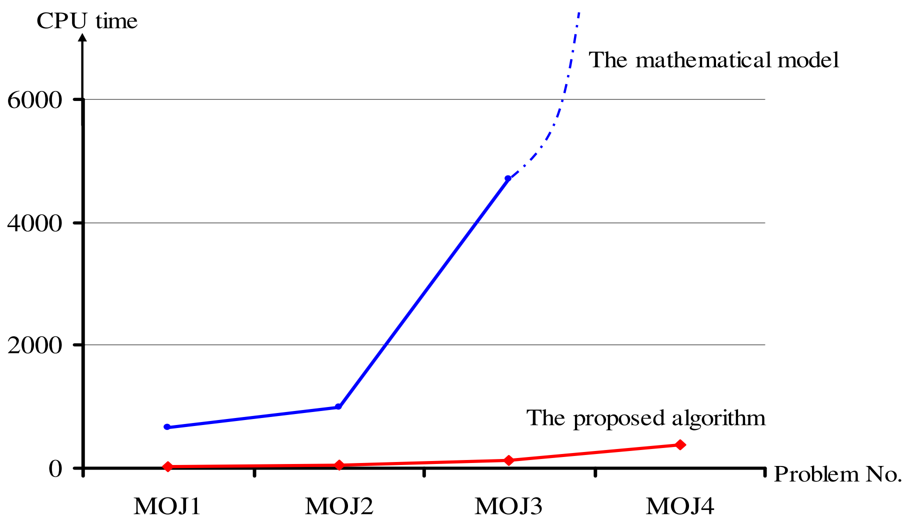

Figure 2. The comparison of CPU time for the proposed algorithm and mathematical model.

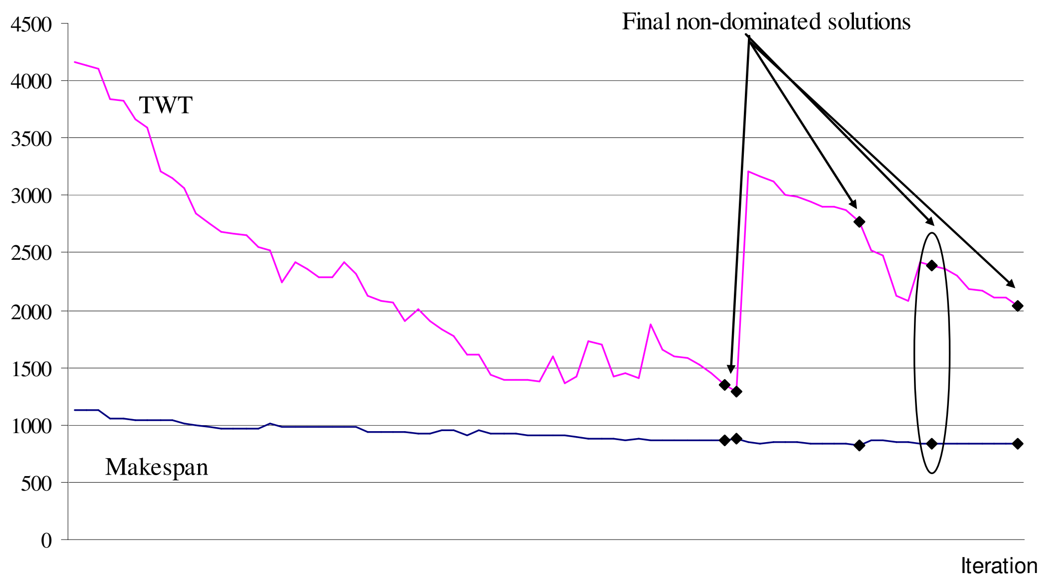

Figure 3. The non-dominated set during the solution process for TWT and $C_{\max}$.

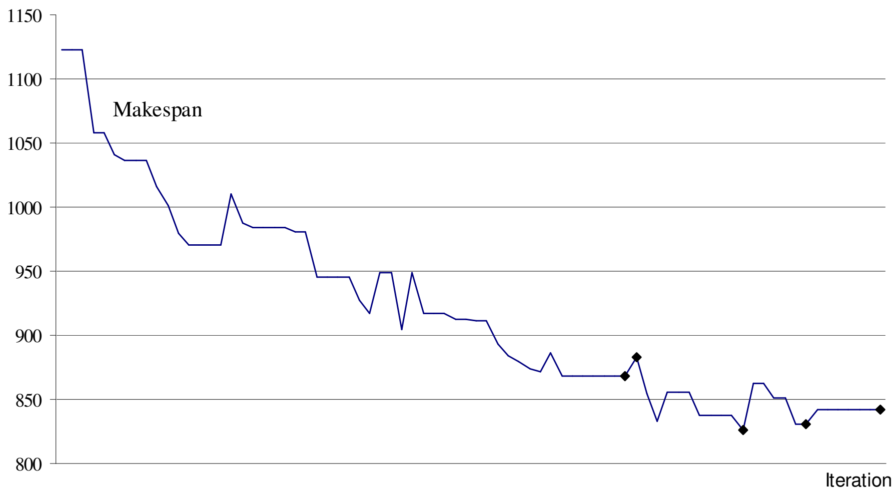

Figure 4. The non-dominated set during the solution process for $C_{\max}$.

References

[1] Asano, M. and Ohta, H., 2002, A heuristic for job shop scheduling to minimize total weighted tardiness. *Computer and industrial engineering,* 42(2-4), 137-147.

[2] Cochran, J. K., Horng, S. M. and Fowler, J. W., 2003, A multi-population genetic algorithm to solve multi-objective scheduling problems for parallel machines. *Computers & Operations Research,* 30(1), 1087-1102.

[3] Drobouchevitch, I. G. and Strusevich, V. A., 2000, Heuristics for the two-stage job shop scheduling problem with a bottleneck machine. *European Journal of Operational Research,* 123(2), 229-240.

[4] Ehrgott, M., and Gandibleux, X., 2000, A survey and annotated bibliography of multi-objective combinatorial optimization. *OR Spektrum,* 22, 425-460.

[5] Fred Choobineh, F., Mohebbi, E. and Khoo, H., 2005, A multi-objective tabu search for a single-machine scheduling problem with sequence-dependent setup times. *European Journal of Operational Research.* Paper in press.

[6] Gawiejnowicz, S., Kurc, W. and Pankowska, L., 2006, Pareto and scalar bicriterion optimization in scheduling deteriorating jobs. *Computers & Operations Research,* 33(3), 746–767.

[7] Golenko Ginzburg, D., and Gonik, A., 2002, Optimal job-shop scheduling with random operations and cost objectives. *International Journal of Production Economics*, 76(2), 147-157.

[8] Hoogeveen, H., 2005, Multi-criteria scheduling. *European journal of operational research,* 167,(3), 592-623.

[9] Kacem, I., Hammadi, S. and Borne, P., 2002, Pareto-optimality approach for flexible job-shop scheduling problems: Hybridization of evolutionary algorithms and fuzzy logic. *Mathematics and Computers in Simulation,* 60, 245-276.

[10] Loukil, T., Teghem, J. and Tuyttens, D., 2005, Solving multi-objective production scheduling problems using metaheuristics. *European journal of operational research.* 161(1), 42-61.

[11] Mattfeld, D. C. and Bierwirth, C., 2004, An efficient genetic algorithm for job shop scheduling with tardiness objectives. *European Journal of Operational Research,* 155(3), 616-630.

[12] Miettinen, K., 1999, *Nonlinear Multi-objective Optimization.* Dordrecht: Kluwer Academic Publishers.

[13] Mosheiov, G. and Oron, D., 2005, A note on flow-shop and job-shop batch scheduling with identical processing-time jobs. *European Journal of Operational Research,* 161(1), 285-291.

[14] Saidi, M. and Fattahi, P., 2006, Flexible job shop scheduling with tabu search algorithm. *Interna-*

tional Journal of Advanced Manufacturing Technology, Paper in Press.

[15] Thiagarajan, S. and Rajendran, C., 2005, Scheduling in dynamic assembly job-shops to minimize the sum of weighted earliness, weighted tardiness and weighted flow time of jobs. *Computers & Industrial Engineering,* 49(4), 463-503.

[16] Ulungu, E. L., Teghem, J., Fortemps, Ph. and Tuyttens, D., 1999, MOSA method: A tool for solving MOCO problems. *Journal of Multi-Criteria Decision Analysis,* 8(4), 221-236.

[17] Varadharajan, T. K. and Rajendran, C., 2005, A multi-objective simulated-annealing algorithm for scheduling in flow shops to minimize the makespan and total flow time of jobs. *European Journal of Operational Research,* 167(3), 772-795.

[18] Wenqia, H. and Aihuaa, Y., 2004, An improved shifting bottleneck procedure for the job shop scheduling problem. *Computers & Operations Research,* 31(1), 2093-2110.

[19] Xia, W. and Wu, Z., 2005, An effective hybrid optimization approach for multi-objective flexible job-shop scheduling problems. *Computers & Industrial Engineering*, 48(2), 409-425.