A Hybrid Multi Objective Algorithm for Flexible Job Shop Scheduling

Parviz Fattahi

Abstract—Scheduling for the flexible job shop is very important in both fields of production management and combinatorial optimization. However, it quit difficult to achieve an optimal solution to this problem with traditional optimization approaches owing to the high computational complexity. The combining of several optimization criteria induces additional complexity and new problems. In this paper, a Pareto approach to solve the multi objective flexible job shop scheduling problems is proposed. The objectives considered are to minimize the overall completion time (makespan) and total weighted tardiness (TWT). An effective simulated annealing algorithm based on the proposed approach is presented to solve multi objective flexible job shop scheduling problem. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the solution process. Numerical examples are used to evaluate and study the performance of the proposed algorithm. The proposed algorithm can be applied easily in real factory conditions and for large size problems. It should thus be useful to both practitioners and researchers.

Keywords—Flexible job shop, Scheduling, Hierarchical approach, simulated annealing, tabu search, multi objective.

I. INTRODUCTION

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, technology 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. The general JSP is strongly NP-hard [1]. In order to match nowadays market requirements, manufacturing systems have to become more flexible and efficient. To achieve these objectives, the systems need not only the automated and flexible machines, but also the flexible scheduling systems. The flexible job shop scheduling problem (FJSP) extends JSP by assuming that, for each given operation, there is at least one instance of the machine type necessary to perform it. The scheduling problem of a FJSP consists of a routing sub-problem, that is assigning each operation to a machine out of a set of capable machines and the scheduling sub-problem, which consists of sequencing the assigned operations on all machines in order to obtain a feasible schedule minimizing a predefined objective function. The FJSP mainly presents two difficulties. The first one is to assign each operation to a machine, and the second one is to schedule these operations in order to make a predefined objective minimal [2]. The FJSP is a much more complex version of the JSP, so the FJSP is strongly NP-hard and combinatorial. It incorporates all of the difficulties and complexities of its predecessor JSP and is more complex than JSP because of the addition need to determine the assignment of operations to machines [3].

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 objective. For example, increasing the input rate of product into a system generally leads to higher throughput, but also to increased work-in-process (WIP) [4].

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

- Review of the multi criteria scheduling problem. Hoogeveen [5] presented a comprehensive review of the published literature on the multi criteria scheduling. He presented that the following performance criteria appeared 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

Parviz Fattahi is with Industrial engineering department, Faculty of Engineering, Bu-Ali Sina University, Hamadan, Iran (e-mail: Fattahi@basu.ac.ir, phone: 00988118257410).

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 [10],[11], flow shop scheduling [12], parallel machine [13] and job shop scheduling [4].

- 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 [7], [10] and job shop scheduling [11].

- 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 [12]-[14], various tardiness objectives [15], [16], penalty cost [17] and various earliness objectives [18]. 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).

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, Meta heuristics, like simulated annealing (SA), Tabu search and genetic algorithms have demonstrated their ability to solve combinatorial problems. So, some authors suggested adapting Meta heuristics in order to solve multi-objective combinatorial problems [20]. In particular Ulungu et al. [21] conceived a multi-objective simulated annealing (MOSA) algorithm for solving combinatorial optimization problems.

The literature of FJSP is considerably sparser than the literature of JSP. Bruker and Schile [19] were among the first to address this problem. They developed a polynomial algorithm for solving the flexible job shop problem with two jobs. For solving the realistic case with more than two jobs, two types of approaches have been used: hierarchical approaches and integrated approaches. In hierarchical approaches assignment of operations to machines and the sequencing of operations on the resources or machines are treated separately, i.e. assignment and sequencing are considered independently, where in integrated approaches, assignment and sequencing are not differentiated. Hierarchical approaches are based on the idea of decomposing the original problem in order to reduce its complexity. This type of approach is natural for FJSP since the routing and the scheduling sub-problem can be separated [3].

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

II. PROBLEM DESCRIPTION AND FORMULATION

A. Flexible Job Shop Scheduling Problem

Flexible job shop scheduling problem (FJSP) 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. The machine set is noted M, $M = \{M_1, M_2,\ldots,M_m\}$. Unless stated otherwise, index i denotes a machine, index j denote jobs and index h denote operations throughout the paper. The execution of each operation h of a job j (noted $O_{j,h}$) requires one machine out of a set of given machines called $M_{j,h} \subset M$ and a process time, $P_{i,j,h}$, for each alternative machine. The set $M_{j,h}$ is defined by $a_{i,j,h}$ as described below. An index k is assigned for each machine that determines the sequence of the assigned operations on it.

B. 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 $x = (x_1, x_2,\ldots,x_n)$ is called a decision vector and $S \subset \mathfrak{R}^n$ 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 $F(x) \in \mathfrak{R}^2$.

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 $f_i(x) \leq f_i(x^*)$ for i=1, 2 and $f_i(x) < f_i(x^*)$ for at least one i [22]. 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 [23].

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.
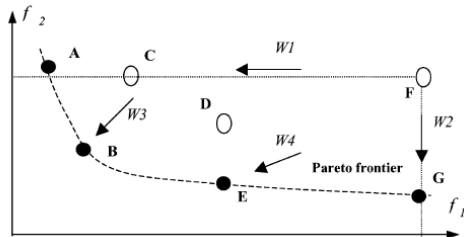

Fig. 1 An illustration example of a non-dominated set

C. Mathematical Model

The model presented by Fattahi, Saidi and Jolai [3] is developed to present a multi objective flexible job shop scheduling problem (MFJSP). Under the assumptions and notations presented in previous sections, the problem is to both determine an assignment and a sequence of the operations on all machines that minimizes the overall completion time (makespan) and total weighted tardiness (TWT) given $n, m, O_{j,h}, a_{i,j,h}, ov_{j,h}$ and $p_{i,j,h}$. The following additional notations are used in the mixed integer linier program formulation of MFJSP with overlapping operations.

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

$$y_{i,j,h} = \begin{cases} 1 & if \quad machine \ i \ selected \ for \ operation \ O_{j,h} \\ 0 & otherwise \end{cases}$$

C_{max} : Makespan time

Ta : Total tardiness of schedule

Ta_j : The tardiness of job j

d_j : The due date of job j

L: A large number.

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

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

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

k_i : The number of assigned operations to machine i.

$ps_{j,h}$: Processing time of operation $O_{j,h}$ after select a machine.

A mixed integer linier program for the FJSP with overlapping operations is then given as

Min C_{max}

Min $Ta = \sum_{j=1}^{n} w_j Ta_j$

s.t.

$$Ta_j \geq \max\left(\left(t_{j,h_j} + ps_{j,h_j}\right) - d_j, 0\right) \quad for \quad j = 1,...n; \quad (1)$$

$$C_{max} \geq t_{j,h_j} + ps_{j,h_j} \quad for \quad j = 1,...,n; \quad (2)$$

$$\sum_i y_{i,j,h} \cdot p_{i,j,h} = ps_{j,h} \quad for \quad j = 1,...,n; h = 1,...,h_j; \quad (3)$$

$$t_{j,h} + ps_{j,h} \cdot ov_{j,h} \leq t_{j,h+1} \quad for \quad j = 1,...,n; \ h = 1,...,h_j - 1; \quad (4)$$

$$t_{j,h} + ps_{j,h} \leq t_{j,h+1} + ps_{j,h+1} - ps_{j,h} \cdot ov_{j,h}$$

$$for \ j = 1,...,n; h = 1,...,h_j - 1 \quad (5)$$

$$Tm_{i,k} + ps_{j,h} \cdot x_{i,j,h,k} \leq Tm_{i,k+1}$$

$$for \ i = 1,..,m; \ j = 1,..,n; h = 1,..,h_j; k = 1,..,k_i - 1; \quad (6)$$

$$Tm_{i,k} \leq t_{j,h} + (1 - x_{i,j,h,k}).L$$

$$for \ i = 1,..,m; \ j = 1,..,n; h = 1,..,h_j; k = 1,..,k_i; \quad (7)$$

$$Tm_{i,k} + (1 - x_{i,j,h,k}).L \geq t_{j,h}$$

$$for \ i = 1,..,m; \ j = 1,..,n; h = 1,..,h_j; k = 1,..,k_i; \quad (8)$$

$$y_{i,j,h} \leq a_{i,j,h} \quad for \ i = 1,..,m; j = 1,..,n; h = 1,..,h_j; \quad (9)$$

$$\sum_j \sum_h x_{i,j,h,k} = 1 \quad for \ i = 1,..m; k = 1,..,k_i; \quad (10)$$

$$\sum_i y_{i,j,h} = 1 \quad for \quad j = 1,..n; h = 1,..,h_j; \tag{11}$$

$$\sum_k x_{i,j,h,k} = y_{i,j,h} \quad for \quad i = 1,...,m; \quad j = 1,...,n; \quad h = 1,...,h_j; \tag{12}$$

$$t_{j,h} \geq 0 \quad for \quad j = 1,\ldots,n; \quad h = 1,\ldots,h_j; \tag{13}$$

$$x_{i,j,h,k} \in \{0,1\} \quad for \quad i = 1,..,m; j = 1,..,n; h = 1,..,h_j; k = 1,..,k_i; \tag{14}$$

$$y_{i,j,h} \in \{0,1\} \quad for \quad i = 1,..,m; j = 1,..,n; h = 1,..,h_j; \tag{15}$$

Constraints (1, 2) determine the two objectives. Constraint (3) determines the processing time of operation $O_{j,h}$ by selected machine. Constraints (4, 5) enforce each job to follow a specified operation sequence and consider the overlapping constraints. Constraint (6) forces each machine to process one operation at a time. Constraints (7) and (8) 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 (9) determines the alternative machines for each operation. Constraint (10) assigns the operations to a machine and sequence assigned operations on all machines. Constraints (11) and (12) force each operation can be performed only on one machine and at one priority. Results of $x_{i,j,h,k}$ yield an assignment each operation on a machine and sequence assigned operations on all machines.

III. THE PROPOSED APPROACH

The scheduling problem of a FJSP consists of a routing sub-problem, that is assigning each operation to a machine out of a set of capable machines and the scheduling sub-problem, which consists of sequencing the assigned operations on all machines in order to obtain a feasible schedule minimizing the predefined objective functions. As discussed previously, in hierarchical approaches assignment of operations to machines and the sequencing of operations on the resources or machines are treated separately, i.e. assignment and sequencing are considered independently. Hierarchical approaches are based on the idea of decomposing the original problem in order to reduce its complexity. In this section, we develop the FJSP algorithm presented by Fattahi, Saidi and Jolai [3] to present a multi objective hybrid algorithm (MOHA) for MFJSP. They compare various hybrid algorithms for FJSP and conclude that the HTSSA algorithm is better than other presented algorithms. HTSSA uses the tabu search algorithm in sequencing algorithm and simulated annealing algorithm in scheduling algorithm based on a hierarchical approach. We develop this algorithm to present a multi objective algorithm which presented in figure2. This algorithm consists of two algorithms: assignment algorithm and scheduling algorithm. The objective functions of each assignment are computed by scheduling algorithm and the initial solution of scheduling algorithm is obtained from the assignment algorithm. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the solution process.

IV. COMPUTATIONAL RESULTS

This section describes the computational experiments which are used to evaluate the effectiveness and performance of the proposed algorithm in finding good quality solutions. For this purpose, we tested the algorithm versus the mathematical model. Although there is a large variety of methods in the literature of multi objective decision making problems to generate the Pareto frontier (such as variants of the parametric method, and the adaptive search method), in this study we use the e-constraint method, since this method is very simple to apply and it is applicable to non convex feasible region problems as well. The e-constraint method can be formulated as follows:
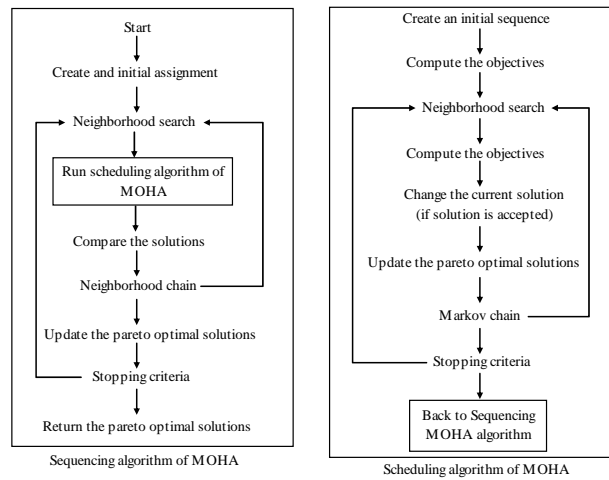

Fig. 2 The proposed MOHA algorithm

$$Min \quad Z_1(x)$$
$$st: \quad Z_2(x) \leq \varepsilon \tag{16}$$
$$x \in X$$

Where ε is a parameter. With varying the value of e systematically, the optimal solutions of this problem give the points of the Pareto frontier. By repeatedly relaxing the upper bound on $Z_2(x)$, and re-optimizing $Z_1(x)$ each time, the points (z_1, z_2) provide the Pareto frontier [24]. This procedure is illustrated in Fig. 3.

To validate the performance of the proposed algorithm, five problems (MOFJ1:2.3.3, MOFJ2:3.2.3, MOFJ3:4.2.3, MOFJ4:3.3.3 & MOFJ5:5.3.5 that are shown in table1, 2, 3, 4 & 5) of multi objective flexible job shop scheduling based on practical data have been selected. These problems are represented by MOFJ (multi objective flexible job shop), n (No. of jobs), h (No. of operations) and m (No. 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 1.80 GHz processor, with 512 Mb

RAM. Their results are shown in Table VI. Moreover, the mathematical model and ε-constraint method is used to evaluate the performance of the MOHA algorithm and its result is presented in Table VII.

A review of the results in Table VI and VII show that, the proposed algorithm is capable to obtain near the optimal solutions. Moreover, the proposed algorithm can obtain all of Pareto solution in a small time. Therefore, the proposed algorithm is useful in multi objective flexible job shop scheduling problems. The non-dominated set for the problem MOJ5 is obtained through the solution process and shown in figure 4. This figure shows that the non-dominated set will be updated during the solution process and the final non-dominated set will be presented.
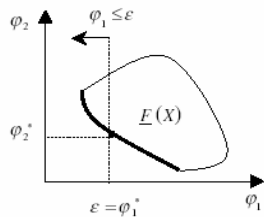

Fig. 3 Illustration of the ε-constraint method [24]

TABLE I
THE DATA OF PROBLEM MOFJ1:2.3.3

Jobs	Due Date	Operation 1		Operation 2		Operation 3	
		Mach. No. *	Pro. time	Mach. No.	Pro. time	Mach. No.	Pro. time
1	200	1	85	2	55	1	95
		2	96	3	48	2	86
						3	125
2	150	1	26	1	54	2	83
		2	48	2	38	3	90

* Mach. No. : Machine number, Pro. time: Process time

TABLE II
THE DATA OF PROBLEM MOFJ2:3.3.3

Jobs	Due Date	Operation 1		Operation 2	
		Mach. No. *	Pro. time	Mach. No.	Pro. time
1	100	1	65	2	45
		2	85	3	56
2	130	1	45	2	54
		2	64	3	68
3	150	1	95	2	90
		3	75	3	80

TABLE III
THE DATA OF PROBLEM MOFJ3:4.2.3

Jobs	Due Date	Operation 1		Operation 2	
		Mach. No. *	Pro. time	Mach. No.	Pro. time
1	150	1	65	1	45
		2	75	3	64
2	110	1	65	2	54
		2	44	3	68
3	170	1	84	2	90
		3	75	3	80
4	150	1	120	2	110
		2	75	3	80

TABLE IV
THE DATA OF PROBLEM MOFJ4:3.3.3

Jobs	Due Date	Operation 1		Operation 2		Operation 3	
		Mach. No. *	Pro. time	Mach. No.	Pro. time	Mach. No.	Pro. time
1	200	1	85	2	55	1	95
		2	96	3	48	2	86
						3	125
2	150	1	26	1	54	2	83
		2	48	2	38	3	90
3	250	1	87	2	90	2	100
		3	60	3	70	3	90

TABLE V
THE DATA OF PROBLEM MOFJ5:5.3.5

Jobs	Due Date	Operation 1		Operation 2		Operation 3	
		Mach. No. *	Pro. time	Mach. No.	Pro. time	Mach. No.	Pro. time
1	300	1	85	3	55	3	95
		2	96	4	88	4	110
						5	125
2	250	1	26	2	54	4	120
		2	48	3	68	5	90
3	150	1	68	3	45	3	87
		2	47	4	56	4	64
						5	120
4	200	2	55	3	66	4	99
		3	48	4	88	5	110
5	250	1	87	2	100	4	110
		3	60	4	120	5	90

TABLE VI
THE FINAL NOM-DOMINATED SETS OBTAINED BY THE PROPOSED ALGORITHM

MOFJ1		MOJ2		MOJ3		MOJ4		MOJ5	
Cmax,	TWT	Cmax,	TWT	Cmax,	TWT	Cmax,	TWT	Cmax,	TWT
245	45	155	60	209	134	235	108	324	155
228	47	164	49	219	98	241	106	303	235
227	53					268	82	316	234
223	92					273	73		
CPU time (s):		CPU time (s):		CPU time (s):		CPU time (s):		CPU time (s):	
30		30		30		30		64	

TABLE VI
THE FINAL NOM-DOMINATED SETS OBTAINED BY THE
e-CONSTRAINT METHOD

MOFJ1		MOJ2		MOJ3		MOJ4	
Cmax,	TWT	Cmax,	TWT	Cmax,	TWT	Cmax,	TWT
245	45	155	60	209	134	235	108
228	47	164	49	219	98	241	106
227	53					268	82
223	92					273	73
Var: 167		Var: 146		Var: 216		Var: 299	
Cons: 308		Cons: 285		Cons: 474		Cons: 612	
CPU time:		CPU time:		CPU time (s):		CPU time (s):	
80		70		3600		7200	

V. CONCLUSION

In this paper, a Pareto approach is proposed to solve the multi objective flexible job shop scheduling problems. The objectives considered are to minimize the overall completion time (makespan) and total weighted tardiness (TWT). An effective simulated annealing algorithm based on the proposed approach is presented to solve multi objective flexible job shop scheduling problem. An external memory of non-dominated solutions is considered to save and update the non-dominated solutions during the solution process. Numerical

experiments show 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. Therefore, the proposed algorithm is useful in multi objective flexible job shop scheduling problems. In this paper two well known objectives are used for the multi objective flexible job shop scheduling problems, so a review on another objectives and methods in this field can be supposed as further research.

REFERENCES

[1] Garey MR, Johnson, DS, Sethi R (1976). The complexity of flow shop and job shop scheduling. Mathematics of Operations Research 1:117-129.

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

[3] Fattahi P., Saidi Mehrabad M., and Jolai F., (2007). Mathematical modeling and heuristic approaches to flexible job shop scheduling problems", Journal of Intelligent Manufacturing, Vol.18:331-342.

[4] Fattahi, P., Saidi Mehrabad, M., Arynezhad M.B. (2006). An algorithm for multi objective job shop problem. Journal of Industrial International, 2, 2.

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

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

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

[8] 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, 772–795.

[9] 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, 1087–1102.

[10] Fred Choobineh, F., Mohebbi, E. and Khoo, H., (2005). A multi-objective tabu search for a single-machine scheduling problem with

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

[12] Droubouchevitch, 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, 229-240.

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

[14] 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, 285–291.

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

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

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

[18] 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, 463–503.

[19] Bruker, P., & Schlie, R. (1990). Job shop scheduling with multi-purpose machines. Computing, 45, 369-375.

[20] Ehrgott, M., and Gandibleux, X., (2000). A survey and annotated bibliography of multi objective combinatorial optimization. OR Spektrum, 22, 425–460.

[21] 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, 221–236.

[22] Miettinen, K., (1999). Nonlinear multi objective optimization. Dordrecht: Kluwer Academic Publishers.

[23] 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.

[24] Ghoseiri, K., Szidarovszky, F., Asgharpour M.J., (2004). A multi-objective train scheduling model and solution. Transportation Research Part B 38: 927–952
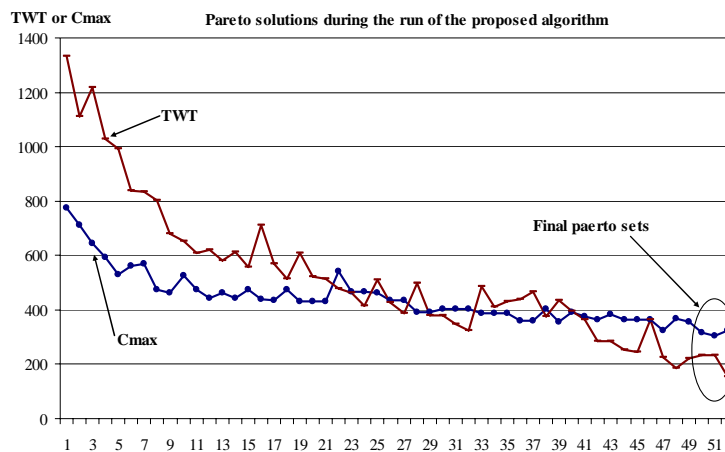
Fig. 4 The Pareto set which is updated during the algorithm for MOFJ5