Archived Simulated Annealing for Unrelated Parallel Machine Scheduling to Minimize Fuzzy Makespan and Tardiness

Wei-Shung Chang[1] Chiuh-Cheng Chyu[†2] Ruei-Chi Li[3]
Department of Industrial Engineering and Management
Yuan-Ze University, Jongli 320, Taiwan
Email: s948905@mail.yzu.edu.tw[1]
iehshsu@saturn.yzu.edu.tw[2]
s975425@mail.yzu.edu.tw[3]

Abstract - *The research proposes a two-phase matching-based decoding scheme that will be incorporated into bi-objective simulated annealing (SA) to solve unrelated parallel machine scheduling problems (UPMSP). The two objectives are to maximize the satisfactions of makespan and average tardiness, in terms of fuzzy measure. The effectiveness of the decoding scheme is investigated using four simulated annealing algorithms: acceptance probability rules based on fitness value difference and Pareto dominance, as well as solution search based on fixed weighted and random weighted directions. An experiment was conducted to evaluate the performance of the proposed eight SAs via instances generated by a method in the literature. The experiment results indicate that (1) two-phase decoding method that uses max-min matching first and Hungarian method second significantly improves proximity quality; (2) Pareto dominance rule probability and random weighted direction search produce solutions with better diversity.*

Keywords: *multi-objective simulated annealing, unrelated parallel machine scheduling, maximum-minimum matching, Hungarian method, fuzzy sets*

1. INTRODUCTION

Scheduling problems are encountered in many production systems. While scheduling jobs, the management generally does not solely focus on one objective. Roy (1985) pointed out that taking several criteria into account enables us to provide the decision maker with a more realistic solution. T'kindt *et al.* (2001) studied a factory manufactures glass bottles the colors of which are selected in advance at the planning phase. Due to manufacturing characteristics and economic concern, two objectives were to optimize workload balance and maximize total profit at the same time. Bertel and Billaut (2004) modeled the processing of checks as a three-stage hybrid flowshop problem with two objectives in lexicographical order: first minimize the maximum tardiness, and then minimize total weighted number of tardy jobs.

This study is focused on unrelated parallel machine scheduling (UPMSP) with two simultaneous maximization objectives – makespan satisfaction and average tardiness satisfaction. The makespan relates to production horizon, and tardiness is relevant to customer delivery service. In this research, a fuzzy measure based on the ratio of area interaction is used to evaluate the satisfaction level of both objectives.

Parallel machine scheduling has been extensively used in many manufacturing environments (Yu *et al.* 2004, Silva and Magalhaes 2006, Wu and Ji 2009, Yang 2009). Most studies regarding UPMSP have focused on one single objective and there have been comparatively few studies on bi-objective UPMSP (Jansen *et al.*, 1999). We refer to Logendran *et al.* (2007) and Allahverdi *et al.* (2008) for a survey of parallel machine scheduling on various objectives and solution methods.

In a real manufacturing environment, the job of processing times and due dates is not constant owing to abnormal machine working or customer demand. Simulated annealing is a compact and robust technique to obtain an optimal solution of a single objective problem and to obtain a Pareto set of solutions for a multi-objective optimization problem (Suman and Kumar, 2006). In this paper, we employed several archived-based multi-objective simulated annealing (A-MOSA; Bandyopadhyay *et al.*, 2008) algorithms to solve UPMSP with two fuzzy maximization objectives – satisfaction grades of makespan and average tardiness. Hereafter, we shall refer to this problem as Fuzzy BIO-UPMSP.

The paper is organized as follows: Section 2 describes the problem. Section 3 illustrates the problem solving

†：Corresponding Author

methods. Section 4 presents the numerical results. Finally, Section 5 concludes the paper.

2. PROBLEM DESCRIPTION

The Fuzzy BIO-UPMSP is defined as follows: (1) the problem contains M parallel machines and J jobs; (2) each machine is allowed to process a job at one time, where processing is non-preemptive; (3) setup times, s_{ijm}, are job sequence-dependent; which are crisp numbers; (4) each job has its own due date, $\tilde{d}_j$, and may also have different processing times, $\tilde{p}_{jm}$, depending on the machine assigned, where both due date and processing time are fuzzy numbers; and (5) two satisfaction grades are maximized: makespan and average tardiness.

Let $\tilde{D} = (D, r)$ be the fuzzy production cycle time. The satisfaction grade on total completion time (makespan) of group G_m under job sequence δ_m is defined as in (1):

$$\tilde{S}_D^{\delta_m}(G_m) = \text{Area}(\tilde{C}_D^{\delta_m}(G_m) \cap \tilde{D})/\text{Area}(\tilde{C}_D^{\delta_m}(G_m)), \quad (1)$$

where $\tilde{C}_D^{\delta_m}(G_m)$ is the fuzzy makespan of G_m under δ_m. The satisfaction grade on tardiness of job j in group G_m under δ_m is defined as in (2):

$$\tilde{S}_d(C_j^{\delta_m}) = \text{Area}(\tilde{C}_j^{\delta_m} \cap \tilde{d}_j)/\text{Area}(\tilde{C}_j^{\delta_m}), \quad (2)$$

where $\tilde{C}_j^{\delta_m}$ is the completion time of job j under δ_m.

Figure 1 presents some example calculations, where LC_j, MC_j, and UC_j denote the lower bound, peak value, and upper bound of the completion time.

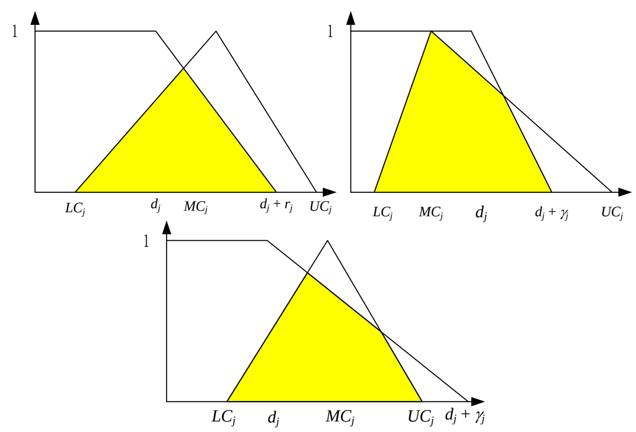

Figure 1: Three possibilities of satisfaction grade of job j's tardiness,

The total satisfaction grade of group G_m under δ_m is

$$\tilde{S}_d^{\delta_m}(G_m) = \sum_{j \in G_m} \tilde{S}_d(C_j^{\delta_m}) \quad (3)$$

The satisfaction grade on makespan for job partition set $\{G_m\}$ is

$$\tilde{S}^*_{C_{\max}}(\{G_m\}) = \underset{m=1,\ldots,M}{Min}\{\underset{\delta_m \in \Delta_m}{Max}(\tilde{S}_D^{\delta_m}(G_m))\}, \quad (4)$$

where Δ_m denotes all possible job sequences on group G_m..

Finally, the satisfaction grade on total tardiness for job partition set $\{G_m\}$ is

$$\tilde{S}^*_d(\{G_m\}) = \sum_{m=1,\ldots,M} \underset{\delta_m \in \Delta_m}{Max}\{\sum_{j \in G_m} \tilde{S}_d(C_j^{\delta_m})\}, \quad (5)$$

3. PROBLEM-SOLVING METHOD

The study presents two simulated annealing algorithms (SA) to solve the proposed BIO-UPMSP. Both multi-objective simulated annealing (MOSA) keep an external archive to maintain a reasonable number of non-dominated solutions during evolution, but they differ in applying acceptance probability rules. One is based on objective fitness (F-MOSA), and the other is based on dominance rule (D-MOSA). The acceptance probability of F-MOSA is determined by comparing the solution quality between the current and the neighborhood solution just generated. On the other hand, D-MOSA determines the probability by the dominance level of the neighborhood solution compared with current solution and the solutions in the archive. The acceptance probability used for D-MOSA refers to Suman (2004, 2005) and Bandyopadhyay *et al.* (2008).

3.1 Encoding and Decoding Schemes

Two encoding schemes are used in F-MOSA and D-MOSA: job – machine list (JML) and job – group list (JGL). To decode an JML with a weighting-objectives vector, the first step is to group the jobs that are assigned to the same machine, and thus there are M single-machine scheduling problems with weighted sum of objectives. Afterwards, a local search using SPT rule to construct an initial solution and followed up by 3-opt refinements is applied to solve each single-machine scheduling problem.

For each group-machine pair, a local optimal bi-objective ($\tilde{S}_D^{\delta_m}(G_m)$, $\tilde{S}_d^{\delta_m}(G_m)$) is obtained. The makespan and total tardiness satisfactions are respectively as follows: $Min\{\tilde{S}_D^{\delta_m}(G_m): m=1,\ldots,M\}$ and $\sum_{m=1,\ldots,M}\left(\tilde{S}_d^{\delta_m}(G_m)\cdot|G_m|\right)\Big/J$

To decode a JGL, a two-phase matching decoding scheme is described in Figure 2, and illustrated through an example with information given in Figures 3(a)-(d).

Decoding procedure with an assigned weighted vector (w, 1-w) on two objectives:

Step 1: grouping the jobs, $\{G_1, \ldots, G_M\}$.

Step 2: for group $k = 1$ to M

for machine $m = 1$ to M,

Apply SPT rule to solve a weighted sum objective single machine scheduling; then use 2-opt or 3-opt to obtain a near optimal solution, $w\cdot \tilde{S}_D^{\delta_m^*}(G_k)+(1-w)\cdot \tilde{S}_d^{\delta_m^*}(G_k)$;

Step 3: Apply max-min matching method to the weighted sum objective M by M matrix obtained in Step 2. Copy the matching cells to the makespan satisfaction matrix and obtain the makespan satisfaction value v^*.

Step 4: Modify the M by M tardiness satisfaction matrix by assigning value 0 to those cells whose makespan satisfaction values are smaller than v^*. Apply the Hungarian method to obtain a matching with maximum total tardiness satisfaction.

Figure 2 Two-phase decoding scheme

An encoding scheme of a 30-job and 5-machine problem: the five groups are G_1 = {1, 5, 10, 11, 19, 28}; G_2 = {2, 18, 24, 25, 29, 30}; G_3 = {4, 6, 9, 12}; G_4 = {2, 7, 8, 14, 17, 21, 26}; G_5 = {13, 15, 16, 20, 22, 23, 27}.

In the example, suppose we take weight 0.4 for makespan satisfaction and weight 0.6 for average tardiness satisfaction. Apply max-min matching we obtain the least weighted sum objective is 0.74, and the corresponding matching set is {(G_1, M_1), (G_2, M_4), (G_3, M_5), (G_4, M_1), (G_5, M_3)}. Next step the procedure turns to makespan satisfaction matrix and using the same matching set to identify the minimum makespan satisfaction v^* = min (0.77, 0.74, 0.76, 0.84, 0.74) = 0.74 (Figure 4(b)). Meanwhile, all cells with $v^* \geq 0.74$ are copied to the average tardiness satisfaction matrix (Figure 4(c)). Acquire the total tardiness satisfaction matrix from 4(d) and apply the Hungarian method to obtain an optimal matching with maximum total tardiness satisfaction, {(G_1, M_1), (G_2, M_5), (G_3, M_2), (G_4, M_2), (G_5, M_3)}. The average tardiness satisfaction in this example is (4.38+4.56+3.03+5.51+5.41)/30 = 0.76. Note that the makespan satisfaction in this new matching set must be no less than the value of v^* = 0.74. By applying the two-phase decoding scheme, we obtain a solution with two objective values of 0.74 for makespan satisfaction and 0.76 for average tardiness satisfaction.

Using the JML decoding scheme for the same example, the matching set will be {(G_1, M_1), (G_2, M_2), (G_3, M_3), (G_4, M_4), (G_5, M_5)}. The makespan satisfaction is min{0.78, 0.78, 0.58, 0.75, 0.60} = 0.58, and the average tardiness satisfaction is (4.38+4.28+2.60+5.37+4.90)/30 = 0.72. Clearly, the two-phase decoding scheme will produce a better solution than JML decoding scheme, but the former will take more computational effort.

group \ machine	$0.4\cdot\tilde{S}_D^{\delta_m^*}(G_k)+0.6\cdot\sum\tilde{S}_d^{\delta_m^*}(G_k)/\lvert G_k\rvert$				
	M_1	M_2	M_3	M_4	M_5
G_1	0.75	0.61	0.79	0.68	0.76
G_2	0.69	0.74	0.58	0.75	0.56
G_3	0.53	0.76	0.62	0.62	0.75
G_4	0.80	0.78	0.72	0.76	0.71
G_5	0.71	0.69	0.76	0.73	0.66

Figure 3(a) Max-min matching on weighted sum with w = 0.4

group \ machine	$\tilde{S}_D^{\delta_m^*}(G_k)$				
	M_1	M_2	M_3	M_4	M_5
G_1	0.78	0.55	0.77	0.66	0.77
G_2	0.72	0.78	0.60	0.74	0.58
G_3	0.50	0.76	0.58	0.60	0.74
G_4	0.84	0.77	0.71	0.75	0.75
G_5	0.70	0.65	0.74	0.72	0.60

Figure 3(b) Qualified cells with least makespan satisfaction v^* = 0.74

	group \ machine	$\sum\tilde{S}_d^{\delta_m^*}(G_k)/\lvert G_k\rvert$				
		M_1	M_2	M_3	M_4	M_5
6	G_1	0.73	0.65	0.80	0.69	0.75
6	G_2	0.67	0.71	0.57	0.76	0.55
4	G_3	0.55	0.76	0.65	0.63	0.76
7	G_4	0.77	0.79	0.73	0.77	0.68
7	G_5	0.72	0.72	0.77	0.74	0.70

Figure 3(c) Identify the cells with $v^* \geq 0.74$ in (b)

group \ machine	$\sum\tilde{S}_d^{\delta_m^*}(G_k)$				
	M_1	M_2	M_3	M_4	M_5
G_1	4.38	0	4.82	0	4.52
G_2	0	4.28	0	4.56	0
G_3	0	3.04	0	0	3.03
G_4	5.39	5.51	0	5.37	4.78
G_5	0	0	5.41	0	0

Figure 3(d) Hungarian method for total tardiness satisfaction

3.2 Neighborhood Solution

Let (f_1, f_2) be two objective values of current solution (CS) with encoding scheme $\{G_1, \ldots, G_M\}$. A neighborhood solution is defined as follows:

Given a weighted vector (w, 1-w), select two groups to swap jobs. One group has worst weighted sum objective values, and the other is randomly selected from the remaining group. Each group randomly selects an integral number from [1, 0.25*J/M] and make an exchange of jobs based on the integral numbers.

3.3 Algorithm F-MOSA

F-MOSA uses objective fitness to determine acceptance probability of a neighborhood solution. If a new neighborhood solution (NS) dominates $h \geq 1$ solutions in the archive, then place NS into the archive and remove the h solutions from the archive. Let (g_1, g_2) be two objective values of NS, $\Delta s_i = f_i - g_i$, i = 1, 2. The acceptance probability $p = \min(1, \exp(-\Delta s_1/T), \exp(-\Delta s_2/T))$, and its initial temperature T_0 is determined based on a K-trial experiment described as follows:

A simple calculation leads to $ln(p) = -(\Delta z_1^k + \Delta z_2^k)/T$. Let $\{(\Delta z_1^1, \Delta z_2^1), \ldots, (\Delta z_1^K, \Delta z_2^K)\}$ be the results of k + 1 consecutive local search, $\Delta \bar{y} = (\max\{0, \Delta z_1^1 + \Delta z_2^1\} + \ldots + \max\{0, \Delta z_1^{K-1} + \Delta z_2^K\})/K$, and T_0 can be obtained by $p_0 = \exp(-\Delta \bar{y}/T_0)$. In F-MOSA, temperature level decreases geometrically.

3.4 Algorithm D-MOSA

The D-MOSA basically follows Bandyopadhyay *et al*. (2008) with slight modification on acceptance probability formula. Let $\Delta dom_{i,j} = (|f_1 - g_1| + |f_2 - g_2|)/2$. There are three cases for determining the acceptance probability of NS and associated updating on the archive.

Case 1: If Current solution (CS) dominates NS, let m be the number of solutions that dominates NS in the archive. Define $\Delta dom_{avg} = \sum_{i=1}^{m} \Delta dom_{i,NS}$ ($\sum_{i=1}^{m} \Delta dom_{i,NS} + \Delta dom_{NS,CS} \Big)/(m+1)$, and acceptance probability of $NS = 1/\left(1+\exp(\Delta dom_{avg} \cdot T_i)\right)$ (6)

Case 2: If CS and NS are not dominated to each other, define $\Delta dom_{avg} = \left(\sum_{i=1}^{m} \Delta dom_{i,NS}\right)/m$; acceptance probability of NS is the same as (6). Update archive by taking into account NS.

Case 3: If NS dominates CS, there are two cases.

Case 3.1: If NS is dominated by at least one solution in the archive, compute the number of solutions that dominate NS and CS respectively in the archive, and denote the minimum of these two numbers as Δdom_{min}. Set the acceptance probability of NS as $1/(1+\exp(\Delta dom_{\min} \cdot T_i))$

Case 3.2: If NS is non-dominated to solutions in the archive, then set the acceptance probability of NS to one and update the archive by NS.

3.5 Search Direction

The aim of our presented algorithms is not to determine a single final solution, but rather to find a sufficiently large number of widely distributed potential efficient solutions for Fuzzy BIO-UPMSP. In order to achieve this goal, two search direction strategies were adopted: (1) FW (fixed weights); (2) RW (random weights).

For algorithms using FW, seven weight vectors were selected: $\{(w_1, w_2) \mid (1.0, 0), (0.8, 0.2), (0.6, 0.4), (0.5, 0.5), (0.4, 0.6), (0.2, 0.8), \text{and } (0, 1.0)\}$. Each algorithm solves the Fuzzy BIO-UPMSP with one weight vector.

In applying algorithms with RW, when an initial or neighborhood solution is produced, the decoding scheme will randomly generate a weight vector from interval [0,1] with sum equal to 1. This weight vector is then used to decode the new neighborhood solution. For F-MOSA and D-MOSA, the number of solutions for termination is set to be the same.

4. NUMERICAL ANALYSIS

This paper presents the results of eight MOSA algorithms. Let F denote F-MOSA, D denote D-MOSA, JM denote JML decoding scheme, M2 denote the two-phase decoding scheme.

4.1 Data Set Generation

Three test instances of problem size, 200 (jobs) x 10 (machines), were generated according to Lee *et al.* (1997) and Saidi *et al.* (2009). The fuzzy processing time of job j on machine m, $\tilde{p}_{jm}$ = (p_{jm}, α_{jm}, β_{jm}), is generated as follows: p_{jm} is a random number from the interval [50, 150] (the mean processing time of a job is $\bar{p}$ = 100); α_{jm} and β_{jm} are integral random numbers from the interval [0, $0.1 \cdot p_{jm}$]. The estimated makespan is $\hat{C}_{\max} = \bar{p} \cdot \mu$, where μ is the total number of jobs divided by the total number of machines. The common due date of all jobs specified by the producer is a trapezoidal fuzzy number $\tilde{D} = (\hat{C}_{\max}, \gamma)$, where γ is a random number from (0, $0.2 \cdot \hat{C}_{\max}$).

The fuzzy due date $\tilde{d}_j = (d_j, \gamma_j)$ of job j is generated using tardiness factors τ and due date range factor R (Lee *et al.* 1997). The higher the τ value, the tighter the tardiness; the larger the R value, the wider spread the range. The value of γ_j is a random number from the interval [0, $0.2 \cdot d_j$]. An experiment was conducted on a problem instance of size 200 x 10, with three parameter vectors (τ, R) = {(0.3, 0.8), (0.5, 0.5), and (0.7, 0.2)}.

4.2 Algorithm Parameters

For both MOSAs, relevant parameters are set to the follow values based on experiments: T_0 = 1.421, α = 0.8, number of decreases = 24, number of neighborhood solutions at each temperature level = 42, and each neighborhood solution executes 30 times of 2-swap or 3-opt to obtain new single-machine schedules for the two changed groups. If a group contains 10 jobs or less, 2-swap is applied; otherwise, 3-opt is used. The termination condition is set to 10,080 solutions calculated. For FW search strategy, each of the seven weighted vectors will calculate 1,440 solutions. For RW search strategy, the SA will restart seven times and each time computes 1,440 solutions using the same parameters setting as FW.

An experiment was conducted to investigate the performance of proposed algorithms. All algorithms were coded in Visual Studio C++.Net 2008 and implemented on PC with Intel core dual 1.8GHz and 2 GB DDRII 566.

4.3 Performance evaluation

Three metrics are used to evaluate the performance of the proposed algorithms: Modified proximity distance (MPD), Hypervolume (Zitzler and Thiele, 1998), and Pareto front rank occupancy (PFRC). The MPD of an algorithm A is defined as follows:

Let Q_A be the set of non-dominated solutions generated by algorithm A, and Q^* be the set of reference Pareto optimal solutions. If $|Q_A| \geq |Q^*|$, for each solution q^* in Q^* find the minimum distance of q^* to Q_A, and then take the average of these distance values. On the other hand, if $|Q^*| \geq |Q_A|$, then the calculation is made by exchanging the roles of Q^* and Q_A.

Hypervolume (HV) can measure both proximity and diversity of a set of non-dominated solutions. PFRC = {PFL_1, PFL_2, …} measures the distribution of Q_A in different Pareto front ranking levels, where PFL_k represents the k_{th} best Pareto front, k = 1, …

Tables 1 to 3 display the performance of various algorithms on the instance with three parameter vectors (τ, R) = {(0.3, 0.8), (0.5, 0.5), and (0.7, 0.2)}. In each instance, the capital letter (a), (b), (c), (d), (e), (f), (g) and (h) denotes the algorithm of D_JM_RW, D_M2_RW, F_JM_RW, F_M2_RW, D_JM_FW, D_M2_FW, F_JM_FW, F_M2_FW, respectively. Moreover, all algorithms with two-phase decoding scheme (M2) outperform those with JML in all measures. F-MOSA with M2 outperforms D-MOSA with M2, but the result is reversed for JML. Algorithms using JML decoding scheme run faster but their performances are worse than algorithms with M2.

Table 1 Algorithm performances for (τ, R) = (0.3, 0.8)

Alg.	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	
$\|E\|$	3	4	1	1	7	6	1	4	
HV (%)	7.5	82.8	5.8	83.4	9.9	81.4	5.8	85	$\|PFL_i\|$
MPD (%)	16.4	4.2	22.8	1.6	10.9	2.3	22.7	0	
PFL_1	0	0	0	0	0	0	0	4	4
PFL_2	0	0	0	1	0	0	0	0	1
PFL_3	0	4	0	0	0	0	0	0	4
PFL_4	0	0	0	0	0	6	0	0	6
PFL_5	0	0	0	0	6	0	1	0	7
PFL_6	2	0	1	0	0	0	0	0	3
PFL_7	1	0	0	0	0	0	0	0	1
$>PFL_8$	0	0	0	0	1	0	0	0	1
CPU time(s)	1.1	55.2	1.1	55.3	1.1	54.8	1.1	55.3	

Table 2 Algorithm performances for (τ, R) = (0.5, 0.5)

Alg.	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	
$\|E\|$	1	2	1	4	1	3	1	3	
HV (%)	4.2	15.8	4.3	66.7	4.1	52.4	4.3	65.7	$\|PFL_i\|$
MPD (%)	22.1	8.9	21.2	0	22.9	3.6	21.2	6.7	
PFL_1	0	0	0	4	0	0	0	0	4
PFL_2	0	0	0	0	0	0	0	3	3
PFL_3	0	0	0	0	0	3	0	0	3
PFL_4	0	2	0	0	0	0	0	0	2
PFL_5	0	0	1	0	0	0	0	0	1
PFL_6	0	0	0	0	0	0	1	0	1
PFL_7	1	0	0	0	0	0	0	0	1
$>PFL_8$	0	0	0	0	1	0	0	0	1
CPU time(s)	1.1	55.4	1.1	54.9	1.1	55.0	1.1	54.9	

Table 3 Algorithm performances for (τ, R) = (0.7, 0.2)

Alg.	(a)	(b)	(c)	(d)	(e)	(f)	(g)	(h)	
$\|E\|$	7	3	2	2	3	4	1	2	
HV (%)	11.9	40.4	3.4	45	4.3	42.8	3.1	45.5	$\|PFL_i\|$
MPD (%)	25.4	33.2	55.4	28.2	44.8	20.1	78.3	0	
PFL_1	0	0	0	0	0	0	0	2	2
PFL_2	0	1	0	2	0	0	0	0	3
PFL_3	0	0	0	0	0	4	0	0	4
PFL_4	0	2	0	0	0	0	0	0	2
PFL_5	6	0	0	0	0	0	1	0	7
PFL_6	0	0	0	0	3	0	0	0	3
PFL_7	0	0	2	0	0	0	0	0	2
$>PFL_8$	1	0	0	0	0	0	0	0	1
CPU time(s)	1.0	56.0	1.0	55.5	1.0	55.4	1.1	54.6	

5. CONCLUSIONS

Over the years, researchers have focused on machine scheduling problems with single objective. In practice, the goal of management is often multi-fold, and the decision maker prefers several quality alternatives for consideration. This research presents several archived MOSAs to solve the UPMSP with two fuzzy objectives: total completion time and total tardiness. Experimental results indicate that the proposed two-phase decoding method can significantly improve the solutions.

ACKNOWLEDGEMENTS

This research was supported by the National Science Council in Taiwan under grant NSC 98-2221-E-155-039.

REFERENCES

Allahverdi, A., Ng, C.T., Ceng, T.C.E. and Kovalyov, M.Y. (2008) A survey of scheduling problems with setup times or costs. *European Journal of Operational Research*, 187, 985-1032.

Bandyopadhyay, S., Saha, S., Maulik, U. and Deb, K. (2008) A Simulated Annealing-Based Multiobjective Optimization Algorithm: AMOSA. *IEEE Transactions on Evolutionary Computation*, 12 (3), 269-283.

Bertel, S. and Billaut, J.-C. (2004) A genetic algorithm for an industrial multiprocessor flow shop scheduling problem with recirculation. *European Journal of Operational Research*, 159, 651-662.

Jansen, K. and Porkolab, J. (1999) Improved approximation schemes for scheduling unrelated parallel-machines. *ACM Symposium on Theory of Computing*, 408-417.

Lee, Y.H. and Pinedo, M. (1997) Scheduling jobs on parallel machines with sequence-dependent setup times. *European Journal Of Operational Research*, 100, 464-474.

Logendran, R., McDonell, B., and Smucker, B., (2007) Scheduling unrelated parallel machines with sequence-dependent setups. *Computers and Operations Research*, 34 (11), 3420-3438.

Roy, B. (1985) *Methodologie multicritére d'aide á la décision* (in French), Economica, Paris.

Saidi, M.M. and Pahlavani, A. (2009) A fuzzy multi-objective programming for scheduling of weighted jobs on a single machine. *International Journal of Advanced Manufacturing Technology*, 45, 122-139.

Silva, C. and Magalhaes, J.M. (2006) Heuristic lot size scheduling on unrelated parallel machines with applications in the textile industry. *Computers and Industrial Engineering*, 50, 76-89.

Suman B. (2004) Study of simulated annealing based multiobjective algorithm for multiobjective optimization of a constrained problem. *Computers and Chemical Engineering*, 28, 1849-1871.

Suman, B. (2005) Self-stopping PDMOSA and performance measure in simulated annealing based multiobjective optimization algorithms. *Computers and Chemical Engineering*, 29, 1131-1147.

Suman, A. and Kumar, P. (2006) A survey of simulated annealing as a tool for single and multiobjective optimization. *Journal of the Operational Research Society*, 57, 1143-1160.

T'kindt, V., Billaut, J.-C., and Prouse, C. (2001) Solving a bicriteria scheduling problem on unrelated parallel machines occurring in the glass bottle industry. *European Journal of Operational Research*, 135 (1), 42-49.

Wu, Y. and Ji, P. (2009) A scheduling problem for PCB assembly-A case with multiple lines. *International Journal of Advanced Manufacturing Technology*, 43, 1189-1201.

Yang, D. (2009) An evolutionary simulation-optimization approach in solving parallel-machine scheduling problem – A case study. *Computers and Industrial Engineering*, 56, 1126-1136.

Yu, L., Shih, H.M., Pfund, M., Carlyle, W.M. and Fowler, J.W. (2004) Scheduling of unrelated parallel machines- An application to PWB manufacturing. *IIE Transactions*, 34 (11), 921-931.

Zitzler, E. and Thiele, L. (1998) Multiobjective optimization using evolutionary algorithms – A comparative case study. *In Parallel Problem Solving from Nature V (PPSN-V)*, 292-301.

AUTHOR BIOGRAPHIES

Wei-Shung Chang is a Ph.D. candidate in the Department of Industrial Engineering and Management, Yuan-Ze University, Taiwan. His research interests include Supply Chain Management, PCB Assembly, and Production Planning and Multi-objective scheduling. His email address is <s948905@mail.yzu.edu.tw>

Chiuh-Cheng Chyu is an associate professor in the Department of Industrial Engineering and Management at Yuan-Ze University in Taiwan. His research interests include Applied Operations Research and Bayesian Decision Analysis. His email address is

<iehshsu@saturn.yzu.edu.tw>

Ruei-Chi Li is a graduate student in the Department of Industrial Engineering and Management, Yuan-Ze University, Taiwan. Her email address is <s975425@mail.yzu.edu.tw>