J. Software Engineering & Applications, **2009, 2: 323-329**
doi:10.4236/jsea.2009.25042 Published Online December 2009 (http://www.SciRP.org/journal/jsea)

A Multi-Criteria Decision Making for the Unrelated Parallel Machines Scheduling Problem

Wei-Shung CHANG, Chiuh-Cheng CHYU

Department of Industrial Engineering and Management, Yuan-Ze University, Taiwan, China.
Email: s948905@mail.yzu.edu.tw, iehshsu@saturn.yzu.edu.tw

Received August 6th, 2009; revised August 31st, 2009; accepted September 7th, 2009.

ABSTRACT

In this paper, we propose a multi-criteria machine-schedules decision making method that can be applied to a production environment involving several unrelated parallel machines and we will focus on three objectives: minimizing makespan, total flow time, and total number of tardy jobs. The decision making method consists of three phases. In the first phase, a mathematical model of a single machine scheduling problem, of which the objective is a weighted sum of the three objectives, is constructed. Such a model will be repeatedly solved by the CPLEX in the proposed Multi-Objective Simulated Annealing (MOSA) algorithm. In the second phase, the MOSA that integrates job clustering method, job group scheduling method, and job group – machine assignment method, is employed to obtain a set of non-dominated group schedules. During this phase, CPLEX software and the bipartite weighted matching algorithm are used repeatedly as parts of the MOSA algorithm. In the last phase, the technique of data envelopment analysis is applied to determine the most preferable schedule. A practical example is then presented in order to demonstrate the applicability of the proposed decision making method.

Keywords: *Multi-Objective Optimization, Unrelated Parallel Machines Scheduling, Simulated Annealing Algorithm, Integer Programming Models, Multi-Criteria Decision Making*

1. Introduction

Parallel machine scheduling problems have been extensively studied in the literature and widely used in many manufacturing environments, such as the drilling operation in a PWB line [1] and glass etch polishing process in the TFT-LCD manufacture. In many real-life situations, the used machines are not always identical in performance. They are different because they were purchased at different times or for different considerations. Some machines may spend much more time on a particular job than others because of their age or design. Consequently, the layout of unrelated parallel machines is more common than identical parallel machines in real manufacturing environments. The unrelated parallel machine scheduling problem (UPMSP) is more difficult than the identical case. Since the latter belongs to NP hard [2], the UPMSP is also NP hard. For further knowledge and recent findings regarding the UPMSP, we refer to [3–4].

The problem solving approach to the UPMSP can be classified into two categories: metaheuristics and exact solution method. In metaheuristics, Hariri and Potts [5] proposed a two-phase method to solve the UPMSP with the objective of minimizing makespan (C_{max}), where the first phase applies an integer programming technique and the second uses the earliest completion time rule to complete the final schedule of the UPMSP. Weng *et al*. [6] proposed seven heuristics for the UPMSP with job sequence dependent setup times and the objective of minimizing the weighted mean flow time. Two priority rules, shortest processing time (SPT) and the minimum sum of setup time and processing time, are respectively employed in the heuristics. The numerical results indicated that their algorithms are capable of finding quality solutions to problems involving 120 jobs with 20 machines in short computational times. Bank and Werner [7] developed a constructive and iterative algorithm to solve the UPMSP with time window constraints on the job release dates and with the objective of minimizing the total weighted lateness.

Glass *et al*. [8] developed a genetic algorithm (GA), simulated annealing (SA), and tabu search (TS) to solve the UPMSP without the sequence dependent setup time constraints. Their experiments conclude that GA performs no better than the other two algorithms. Sirvastava [9] proposed a TS algorithm that could find high quality solutions in a short time for a part of the same instances.

Kim *et al*. [10–11] proposed an SA to solve the

UPMSP with a goal of total tardiness minimization, while taking into consideration job sequence-dependent set up times. Logendran *et al.* [4,12] developed a TS for the same problem with additional considerations of dynamic release dates and time window machine availability, where the objective was to minimize the sum of weighted tardiness jobs. Chen [13] presented a record-to-record algorithm with tabu list to solve the UPMSP with the goal of minimizing the maximum tardiness. This paper also presented a threshold accepting algorithm with tabu list to solve the UPMSP to minimize the total tardiness.

The branch and bound (B&B) methods are commonly used to optimally solve the UPMSP in the literature [14–18].

Most research on the UPMSP has been focused on a single objective only and there have been comparatively fewer studies on the multi-objective UPMSP. Suresh *et al.* [19] developed a TS for UPMSP with two objectives: minimizing the maximum makespan (C_{max}) and the maximum tardiness. The tabu list keeps the record of newly found non-dominated solutions. Jansen *et al.* [20] modified the TS by Suresh *et al.* and solve the UPMSP with the objectives of minimizing C_{max} and cost of scheduling. In this paper, a simulated annealing that interacts with the commercial software package CPLEX is developed for solving the UPMSP with three objectives – minimizing $C_{\max}$, total flow time, and total number of tardy jobs. Since only one schedule will be implemented in a real situation, a decision procedure is suggested to make the most preferable choice of the candidate solutions.

Simulated annealing (SA) has been known as a compact and robust technique to solve many NP-hard problems, including both single objective and multi-objective ones. It can provide excellent solutions to these problems with a substantial reduction in computational time. SA was first introduced by [21]. We refer to [22–24] for surveys on single objective SA, and [25] for surveys on multi-objective SA.

The remainder of this paper is structured as follows. Section 2 describes the problem under study. Section 3 presents the proposed solution approach. Section 4 presents the numerical results from the proposed method used to solve a problem with real life data. Finally, Section 5 concludes the paper with implications for future research.

2. Problem Description

The aim of this study is to develop a systematic solution method for determining the most preferred schedule among a non-dominated set of schedules found by the proposed hybrid simulated annealing algorithm. In this section, first of all, notations used in this paper are introduced; secondly, the studied problem is described, including a multi-objective mathematical model.

2.1 Notations

Symbols:

i: machine index, $i = 1,\dots, M$

j: job index, $j = 1,\dots, J$

M: total number of machines used

J: total number of jobs to be processed

p_{ij}: processing time of job j performed by machine i

s_{ij}: set up time for job j on machine i

d_j: due date of job j

G_m: set of jobs in group m, $m = 1,\dots, M$

Decision variables:

C_{max}: maximum makespan (completion time) among all machines

$C^i_{\max}$: makespan of machine i

C_{ij}: completion time of job j on machine i

T_{ij}: tardiness of job j on machine i, $T_{ij} = \max\{0, C_{ij} - d_j\}$

U_{ij}: tardiness state of job j if it is processed on machine i: value is 1 if tardy;otherwise value is 0; number of tardy jobs for machine i

y_{ij}: value is 1 if job j is assigned to machine i; value is 0 otherwise

x_{ijk}: value is 1 if both jobs j and k are assigned to machine i and job j immediately precedes job k; value is 0 otherwise

2.2 Problem Definition

We consider the following manufacturing environment: there are I different machines in parallel with a total number of J jobs to be processed, and a job refers to a customer's order. The problem under study assumes that each job may have different processing times depending on the assigned machine, each machine will process one job at a time, and the processing is non-preemptive. The setup time of each job is assumed to be machine dependent but not job sequence dependent; thus, the setup time is included in the processing time. The scheduling problem considered in this study aims to minimize three objectives simultaneously: C_{max}, total flow time, and total number of tardy jobs.

2.2.1 Mathematical Formulation

$$Minimize\ \ (C_{\max},\ \sum_{i=1}^{M}\sum_{j=1}^{J} C_{ij},\ \sum_{i=1}^{M}\sum_{j=1}^{J} U_{ij}) \tag{1}$$

$$s.t.\ \ C_{\max} \ge C^i_{\max} \quad \forall\ i \tag{2}$$

$$C^i_{\max} \ge C_{ij} \quad \forall\ i, j \tag{3}$$

$$C^i_{\max} = \sum_{j=1}^{J} p_{ij} \cdot y_{ij} \quad \forall i \tag{4}$$

$$T_{ij} \ge 0;\ T_{ij} \ge C_{ij} - d_j \quad \forall i, j \tag{5}$$

$$T_{ij} \le M_{big} \cdot U_{ij} \quad \forall i, j \tag{6}$$

$$C_{ik} - p_{ik} + M_{big} \cdot (1 - x_{ijk}) \ge C_{ij} \quad \forall i, j, k \tag{7}$$

$$\sum_{k=1, k \ne j}^{J} x_{ijk} = y_{ij} \quad \forall i, j \tag{8}$$

$$C_{ij} \ge 0 \quad \forall i, j \,,$$

$$y_{ij}, U_{ij} = 0,1 \quad \forall i, j; \quad x_{ijk} = 0,1 \quad \forall\, i, j, k \tag{9}$$

where M_{big} is a big number.

In the model, Equation (1) presents the three objective functions of the UPMSP; inequality set (2) shows the first objective is to minimize the maximin makespan of all machines; constraint set (3) specifies the makespan of each machine; constraint set (4) defines the makespan of each machine; constraint set (5) defines the tardiness of each job; constraint sets (5) and (6) together define the number of tardy jobs; constraint set (7) specifies the starting time relationships between jobs under a certain processing sequence in the same group. Equation set (8) ensures that each individual job will be processed by only one machine.

2.2.2 Multi-Criteria Decision Making

Given a set of non-dominated schedules (or alternatives) to the problem under study, we seek to select the most preferable one. Various approaches can be applied to solve this decision problem. Some well known methods are as follows: 1) construct a multi-attribute utility function [26] which is defined on the objective space, and then take the alternative with the maximum utility value; 2) assign priorities to objectives and take the alternative with the best Lexicographic order; 3) select the alternative that has the maximum AP efficiency ratio in data development analysis [27] or that has the maximum score using principle component analysis [28].

In the case that there are too many non-dominated solutions, a simple three-stage method will be adopted: 1) select a reasonable number of diversified solutions; 2) decide what the inputs and the outputs should be; 3) compute the AP efficiency ratio and choose the one which ranks first.

3. Problem Solving Method

In this paper, we developed a problem solving method that uses the simulated annealing (SA) algorithm as the main framework and employs the CPLEX optimizer to solve the multi-objective scheduling problem with an assigned weight vector. The acceptance probability takes into account the upgrading or downgrading of each individual objective function at each step of generating a neighborhood solution, and is defined as the product of each individual acceptance probability with respect to the change of each objective at each step.

In order to produce an acceptable number of non-dominated schedules, we adopt the following strategy: 1) predetermine a set of weight vectors for the three- objective scheduling problem; 2) solve optimally or near optimally the multi-objective scheduling problem with the target of minimizing the weighted sum of the objectives for each weight vector; 3) find higher quality diversified solutions using different initial solution. In the study, three priority rules, EDD (early due date), SPT (shortest processing time), and CR (critical ratio) are used to generate an initial solution.

3.1 Encoding and Decoding Scheme

In the proposed SA, a job list is used to represent a schedule of UPMSP. A sublist m corresponds to the set of jobs assigned to group m. To generate an initial job list, all jobs are first sorted first based on a priority rule, and then placed one by one into each sublist according to the order of the list. In doing so, a set of job groups is formed, with the number of groups equal to the number of machines. Given a job list, a neighborhood solution will be generated by performing a 3-opt operation on the list.

Figure 1 illustrates the decoding process using 15 jobs and four machines. First, a weight vector (λ_1, λ_2, λ_3) is given for the three objectives In step 1, 15 jobs are grouped and sequenced according to the priority rule. In the example, group 1 (sublist 1) contains jobs 2, 5, 6, and 13. In step 2, based on the results in step 1, we can obtain an initial solution for each group-machine pair and further improve it by single machine scheduling heuristic (SMSH). Note that in this step the three objective values are normalized using Equations (12)–(14). Since there are four machines, a 4 by 4 yielding 16 group-machine pairs are computed. Figure 2 displays the results of step 2.

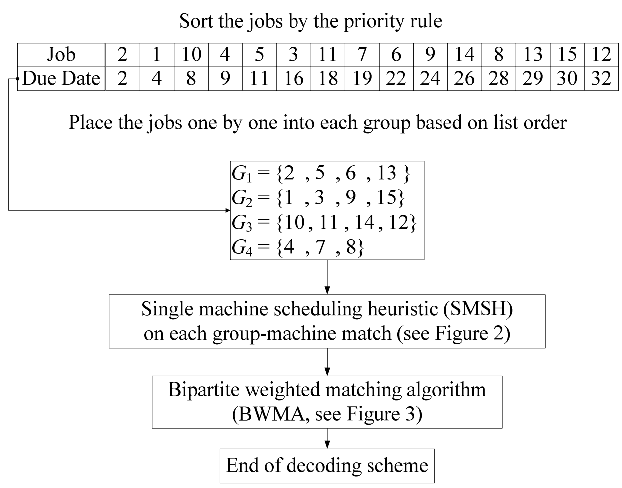

Figure 1. Example of decoding scheme

Single machine scheduling heuristic (SMSH)

Weighted sum objective $= \lambda_1 \cdot f_1 + \lambda_2 \cdot f_2 + \lambda_3 \cdot f_3$

Group 1

Machine	Job sequence	Weighted sum objective
1	2 - 5 - 6 - 13	600
2	6 - 5 - 13 - 2	720
3	2 - 6 - 13 - 5	650
4	5 - 2 - 13 - 6	680

Group 2

Machine	Job sequence	Weighted sum objective
1	15 - 1 - 3 - 9	700
2	15 - 3 - 9 - 1	500
3	3 - 15 - 9 - 1	680
4	9 - 15 - 3 - 1	540

Group 3

Machine	Job sequence	Weighted sum objective
1	10 - 11 - 12 - 14	720
2	11 - 10 - 12 - 14	469
3	10 - 12 - 14 - 11	480
4	10 - 14 - 12 - 10	620

Group 4

Machine	Job sequence	Weighted sum objective
1	4 - 7 - 8	369
2	4 - 8 - 7	450
3	7 - 8 - 4	440
4	4 - 7 - 8	500

Figure 2. Applying SMSH to all group-machine pairs

Bipartite weighted matching algorithm (BWMA)

Group \ Machine	Weighted sum objective 1	2	3	4
G_1	600	720	650	680
G_2	700	500	680	540
G_3	720	469	480	620
G_4	369	450	440	500

Return to three objectives for each match

Assignment	Job sequence	$G_k(C_{max}, \Sigma C_j, \Sigma U_j)$
G_1 - Machine 3	2 - 6 - 13 - 5	G_1(487,900, 2)
G_2 - Machine 4	9 - 15 - 3 - 1	G_2(560,750, 1)
G_3 - Machine 2	11 - 10 - 12 - 14	G_3(475,675, 0)
G_4 - Machine 1	4 - 7 - 8	G_4(520, 720,1)
Total objectives (C_{max}, ΣC_j, ΣU_j)		(560,3045, 4)

Figure 3. Optimal group-machine matching by BWMA

Finally, the bipartite weighted matching algorithm (BW MA) is applied to find the optimal group-machine matching and this concludes the decoding scheme (Figure 3).

3.1.1 Single Machine Scheduling Problem with a Weighted Sum of Objectives

This section presents the mathematical model of the weighted sum objectives of the single machine scheduling problem, which will then be solved by the CPLEX solver in the interactive SA algorithm. Given a weight vector, (λ_1, λ_2, λ_3) with $\lambda_1 + \lambda_2 + \lambda_3 = 1$, $\lambda_1, \lambda_2, \lambda_3 \geq 0$, and group Gm is assigned to machine i, the mathematical model is as follows:

$$Min \quad F_{i,m}(\lambda_1, \lambda_2, \lambda_3) = \lambda_1 \cdot f_1^{i,m} + \lambda_2 \cdot f_2^{i,m} + \lambda_3 \cdot f_3^{i,m}$$

s.t. (3), (4), (5), (6), (7) for $j, k \in G_m$

$$\sum_{j,k \in G_m, j \neq k} x_{ijk} = 1 \quad k \in G_m \tag{10}$$

$$C_{ij} \geq 0 \quad j \in G_m \quad x_{ijk} = 0,1 \quad j,k \in G_m \tag{11}$$

where

$$f_1^{i,m} = (C_{\max}^i - g_{\min}^i) / (g_{\max}^i - g_{\min}^i) \tag{12}$$

$$f_2^{i,m} = (\sum_{j \in G_m} C_j - h_{\min}^i) / (h_{\max}^i - h_{\min}^i) \tag{13}$$

$$f_3^{i,m} = (\sum_{j \in G_m} U_{ij} - u_{\min}^i) / (u_{\max}^i - u_{\min}^i) \tag{14}$$

and ($g_{\min}^i, h_{\min}^i, u_{\min}^i$) and ($g_{\max}^i, h_{max}^i, u_{\max}^i$) are the ideal solution and anti-ideal solution to the three-objective single machine scheduling problem, respectively.

3.1.2 Assignment Problems between Job Groups and Machines

Given an assigned weight vector (λ_1, λ_2, λ_3) and a set of job groups {G_m, m = 1,…, M}, a total of $M \times M$ single machine scheduling solutions can be created, each of which corresponds to a group of jobs being processed by a machine. The problem in this step is a bipartite weighted maching problem and will be solved by the Hungarian method. The mathematical model of the group-machine assignment problem is presented as follows:

$$Min \quad \sum_{m=1}^{M} \sum_{i=1}^{M} F_{i,m}(\lambda_1, \lambda_2, \lambda_3) \cdot y_{im}$$

s.t.

$$\sum_{i=1}^{M} y_{im} = 1 \qquad m = 1, \ldots, M \tag{15}$$

$$\sum_{m=1}^{M} y_{im} = 1 \qquad i = 1, \ldots, M \tag{16}$$

$$y_{im} \geq 0 \qquad m, i = 1, \ldots, M$$

3.2 MOSA Algorithm

The following presents the MOSA algorithm:

1) Select a wide diversified set $\Delta = \{\underline{\lambda}^k : k = 1, \ldots, K\}$, where each $\underline{\lambda}^k$ corresponds to the kth weight vector of the three objectives. Choose a set of priority rules, Pr = {pr^l: l = 1, …, L}, which will subsequently be used to create the initial solutions. Set the number of temperature levels = NTL, and the number of maximum iterations at each temperature level = ITN. k = 1; l = 1, ntl = 1, $T = T_1$, itn = 1.
2) Apply priority rule pr^l to generate an initial partitioned set of jobs, {G_m: m = 1,…, M}; Compute

$F_{i,m}(\lambda_1^k, \lambda_2^k, \lambda_3^k)$; employ the Hungarian algorithm to solve the group jobs – machines assignment problem; evaluate all objective functions and put them into a Pareto set of solutions. Let x = current solution with $Z(x) = (z_1(x), z_2(x), z_3(x))$ be the corresponding non-normalized objective values.

3) Give a random perturbation and generated a new partitioned set of jobs. $\{G_m^{'} : m = 1,\ldots, M\}$; Compute $F_{i,m}(\lambda_1^k, \lambda_2^k, \lambda_3^k)$; employ Hungarian algorithm to solve the group jobs – machines assignment problem and obtain a new solution y; evaluate all objective values: $Z(y) = (z_1(y), z_2(y), z_3(y))$.
4) Compare y with all solutions in the Pareto set and update the Pareto set, if necessary.
5) If y is archived, make it the current solution by putting $x = y$ and move on to step 7; otherwise compute $\Delta z_r = z_r(y) - z_r(x)$, r = 1, 2, 3. Assign acceptance probability to each of objective functions as follows:

   $p_r = 1$ if $\Delta z_r \le 0$; $p_r = (\exp(-\Delta z_r / T))^{\lambda_r^k}$ if $\Delta z_r > 0$, r = 1, 2, 3;

   Acceptance probability = $p_1 \cdot p_2 \cdot p_3$
6) If y is accepted, then set $x = y$;
7) $itn = itn + 1$; if (itn > ITN), set $ntl = ntl + 1$; $T_{ntl} = T_{ntl-1} \cdot \alpha$.
8) If (ntl > NTL), terminate and output the best solution; otherwise, go to step 2.

3.3 Choosing the Best Alternative

In this paper, we will choose the AP efficiency ratio for making the best choice among many competitive alternatives. In the three objectives, a small C_{max} usually implies a high utilization of machine(s), i.e., how much time is taken to complete the production. Furthermore, the sum of the completion times of J jobs gives an indication of the total holding or inventory costs incurred by the schedule [29]. These two objectives will be regarded as the inputs when computing the AP ratio. On the other hand, the number of tardy jobs represents the number of unsatisfied customers. This value will instead be taken as an output. Since a small number of tardy jobs imply a better performance, we use a total number of jobs minus the number of tardy jobs in a schedule as the performance measure to indicate the level of customers' satisfaction of that schedule.

4. Numerical Results

In this section, the computational characteristics and effectiveness of the proposed interactive SA algorithm are evaluated via a practice example, which arises in the glass etch polishing during the Cell manufacturing stage of the TFT-LCD production process. The glass etch polishing operation is independent of the other manufacturing processes, and is required only for some types of TFT-LCD products.

In the example, a collection of real life data for four different machines and 21 different products was made. The data contains the information of the processing time of each job (a batch of the same products) on each individual machine. Our observation indicates that the setup time of a job is approximately the same in each machine, regardless of whichever its preceding job is. But the setup time may be different in different machines. There- fore, it is assumed that the setup time of a job is only machine-dependent rather than job-sequence-dependent, and is included in the processing time.

In the experiment, the SA algorithm was coded in Visual Studio C++. NET and executed on a computer with Intel core dual, 1.8GHz and 2 GB DDR566, and used the CPLEX 10.0 optimizer to solve the weighted sum objective single machine scheduling problem. The parameters setting of the SA is as follows: NTL = 20, ITN = 5, T_1 = 100, and α = 0.95.

Three priority rules are chosen to construct the initial solutions. They are early due date (EDD), shortest process time (SPT), and critical ratio (CR), where CR is defined as the ratio of the due date over the average processing time. Table 1 displays the best solutions found

Table 1. Best solutions for various weight vector with EDD initial solution

λ_1 λ_2 λ_3	C_{max}	Total flow time	No. tardy jobs	CPU (sec.)
0.1, 0.1, 0.8	439	2105.9	2	201.5
* 0.1, 0.2,0.7	344	1831.48	1	196.9
0.1, 0.3, 0.6	556	2276.01	2	200.2
0.1, 0.4, 0.5	367	1888.4	3	187
0.1, 0.5, 0.4	372	1772.98	3	198.1
0.1, 0.6, 0.3	588	2542.73	1	197.5
0.1, 0.7, 0.2	450	1970.37	1	190.7
0.2, 0.1, 0.7	596	2189.48	0	205.4
0.2, 0.2, 0.6	598	2267.12	2	201.1
0.2, 0.3, 0.5	726	2715.26	1	194.5
0.2, 0.4, 0.4	427	2312.18	2	196.9
0.2, 0.5, 0.3	382	1906.4	3	195.9
0.2, 0.6, 0.2	400	2330.29	3	197.5
0.3, 0.1, 0.6	475	2099.12	1	201.0
* 0.3, 0.2,0.5	360	1774.87	2	203.4
0.3, 0.3, 0.4	536	2103.48	1	210.9
0.3, 0.4, 0.3	389	2241.87	3	196.9
0.3, 0.5, 0.2	769	2890.04	1	194.0
0.4, 0.1, 0.5	381	1924.4	1	199.8
0.4, 0.2, 0.4	382	1910.4	2	197.0
0.4, 0.3, 0.3	479	2138.9	1	202.5
* 0.4, 0.4,0.2	527	2019.48	0	200.1
* 0.5, 0.1,0.4	255	1770.4	3	199.8
0.5, 0.2, 0.3	367	2083.79	2	195.6
0.5, 0.3, 0.2	450	2275.62	1	202.3
0.6, 0.1 ,0.3	389	1879.73	2	206.7
0.6, 0.2, 0.2	663	2471.01	1	197.8

Table 2. Non-dominant solutions found in all trials

λ_1 λ_2 λ_3	C_{max}	Total flow time	No. tardy jobs
0.4, 0.4, 0.2	527	2019.48	0
0.3, 0.1, 0.6	304	1794.73	1
0.2, 0.3, 0.5	241	1748.51	2
0.4, 0.4, 0.2	245	1746.51	2

Table 3. Ranks of the four alternatives based on the AP ratio

Alternatives	Input 1	Input 2	Output	CCR	AP ratio
1	527	2019.48	1.000	0.933	0.933
2	304	1794.73	0.952	1.000	1.024
3	241	1748.51	0.905	1.000	1.017
4	245	1746.51	0.905	0.999	0.999

Table 4. Numerical results of ARENA

Dispatching rules	C_{max}	Total flow time	No. tardy jobs
SPT	262	2638	15
EDD	294	3391	20
CR	597	4926	21

with respect to each weight vector and the EDD initial solution. In order to find a limited number of high quality non-dominated solutions, for each weight vector, the SA will output the solution with the minimum weighted objective value.

As observed from Table 1, only four local non-dominated solutions were found with the EDD initial solution. However, there are totally only four global non-dominated solutions found by the SA, as presented in Table 2. The first one was produced with the EDD initial solution, the second with the SPT, and the rest two with the CR. Three local non-dominated solutions in Table 1 were removed when compared with the second and third solutions in Table 2.

To select the most preferable schedule among these four alternatives, the ranking procedure proposed by [26] is applied. When applying the procedure, the values of C_{max} and total flow time are treated as inputs. The value of C_{max} can be viewed as the total time of the equipments used in the production, and the flow time gives an indication of the total holding or inventory costs incurred in the production. The number of tardy jobs occurs in a schedule will be viewed as the output, since it represents the number of customers who will not be satisfied with the purchase service. The following scoring method for this output is adopted:

(total no. of jobs – total no. of tardy jobs) / (total no. of jobs)

Table 3 presents the ranks of the four non-dominated schedules. Both the second and the third schedules have a CCR ratio [30] of one, which implies both are effective. However, a further comparison based on the AP ratio indicates the second schedule should be the most preferable. The proposed decision-making model is more proper if it includes the consideration of the cost of schedules. In unrelated parallel machine scheduling problems, a job may take different processing times on different machines, and thus its processing cost may also be different when processed by different machines.

Table 4 shows the numerical results by applying the simulation software ARENA to the instance using three different dispatching rules commonly used in industry. Clearly, the SPT rule works much better than the other two, but its solution is considerably dominated by the second to fourth solutions in Table 2. The proposed algorithm is superior to the professional software in providing quality solutions for the instance.

5. Conclusions

In this paper, we proposed an interactive simulated annealing algorithm aimed at searching for a set of near Pareto optimal solutions to the unrelated parallel machine scheduling problem with three objectives: minimizing total completion time, total flow time, and total number of tardy jobs. A commercial optimization software package IILOG CPLEX is served as a function in the SA algorithm, with the mission of solving optimally the subproblem - weighted sum objective single machine scheduling problems. To produce an acceptable number of high quality non-dominated solutions, a unique best schedule is found with respect to each weight vector. The ranking procedure proposed by Andersen and Petersen is then applied to select the most preferable schedule, using total completion time and total flow time as the inputs, and total number of tardy jobs as the output.

Further research would include the cost of schedules as one of the inputs in the decision-making process as to select the best schedule. In addition, for problems of large size, hybrid evolutionary algorithms will be more favorable than the current method, in which the CPLEX optimizer occupies a large percentage of computational time. When solving large size problems, the number of non-dominated solutions would be very large. An interesting research direction may be focused on developing a good method for making the best selection among the huge number of near Pareto-optimal solutions.

6. Acknowledgments

This research was partially supported by the National Science Council in Taiwan under grant NSC 95-2221-E-155-045.

REFERENCES

[1] L. Yu, H. M. Shih, M. Pfund, W. M. Carlyle, and J. W.

Fowler, "Scheduling of unrelated parallel machines-An application to PWB manufacturing," IIE Transactions, Vol. 34, No. 11, pp. 921–931, 2004.

[2] M. K. Richard, "Reducibility among combinatorial problems," in R. E. Miller and J. W. Thatcher (editors): Complexity of Computer Computations, pp. 85–103, Plenum, New York, 1972.

[3] M. Pfund, J. W. Fowler, and J.N.D. Gupta, "A survey of algorithms for single and multi-objective unrelated parallel-machine deterministic scheduling problems," Journal of the Chinese Institute of Industrial Engineers, Vol. 21, No. 3, pp. 230–241, 2004.

[4] R. Logendran, B. McDonell, and B. Smucker, "Scheduling unrelated parallel machines with sequence-dependent setups," Computers and Operations Research, Vol. 34, No. 11, pp. 3420–3438, 2007.

[5] M. A. Hariri and C. N. Potts, "Heuristics for sched uling unrelated parallel machines," Computers and Operations Research, Vol. 18, No. 3, pp. 323–331, 1991.

[6] M. Weng, J. Lu, and H. Ren, "Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective," International Journal of Production Economics, Vol. 70, pp. 215–226, 2001.

[7] J. Bank and F. Werner, "Heuristic algorithms for unrelated parallel machine scheduling with a common due date, release dates, and linear earliness and tardiness penalties," Mathematical and Computer Modelling, Vol. 33, pp. 363–383, 2001.

[8] A. Glass, C. N. Potts, and P. Shade, "Unrelated parallel machine scheduling using local search," Mathematical and Computer Modeling, Vol. 20, No. 2, pp. 41–52, 1994.

[9] B. Srivastava, "An effective heuristic for minimizing make- span on unrelated parallel machines," Journal of the Operational Research Society, Vol. 49, pp. 886–894, 1997.

[10] W. Kim, K. H. Kim, W. Jang, and F. F. Chen, "Unrelated parallel machine scheduling with setup times using simulated annealing," Robotics and Computer Integrated Manufacturing, Vol. 18, No. 3-4, pp. 223–231, 2002.

[11] W. Kim, D. G. Na, and F. F. Chen, "Unrelated parallel machine scheduling with setup times and total weighted tardiness objective," Robotics and Computer Integrated Manufacturing, Vol. 19, No.1-2, pp. 173–181, 2003.

[12] R. Logendran and F. Subur, "Unrelated parallel machine scheduling with job splitting," IIE Transactions, Vol. 36, No. 4, pp. 359–72, 2004.

[13] J. F. Chen and T. H. Wu, "Total tardiness minimization on unrelated parallel machine scheduling with auxiliary equipment constraints," Omega-International Journal of Management Science, Vol. 34, pp. 81–89, 2006.

[14] J. F. Chen, "Minimization of maximum tardiness on unrelated parallel machines with process restrictions and setups," International Journal of Advanced Manufacturing Technology, Vol. 29, No. 5, pp. 557–563, 2006.

[15] S. Martello, F. Soumis, and P. Toth, "Exact and approximation algorithms for makespan minimization on unrelated parallel machines," Discrete Applied Mathematics, Vol. 75, pp. 169–188, 1997.

[16] Lancia, "Scheduling jobs with release dates and tails on two unrelated parallel machines to minimize the makespan," European Journal of Operational Research, Vol. 120, pp. 277–288, 2000.

[17] C. F. Liaw, Y. K. Lin, C. Y. Cheng, and M. Chen, "Scheduling unrelated parallel machines to minimize total weighted tardiness," Computers and Operations Research, Vol. 30, pp. 1777–1789, 2003.

[18] P. L. Rocha, M. G. Ravetti, G. R. Mateus, and P. M. Pardalos, "Exact algorithms for a scheduling problem with unrelated parallel machines and sequence and machine-dependent setup times," Computers and Operations Research, Vol. 35, No. 4, pp. 1250–1264, 2008.

[19] V. Suresh and D. Chaudhuri, "Bicriteria scheduling problem for unrelated parallel-machines," Computers and Industrial Engineering, Vol. 30, pp. 77–82, 1996.

[20] K. Jansen and L. Porkolab, "Improved approximation schemes for scheduling unrelated parallel-machines," ACM Symposium on Theory of Computing, pp. 408–417, 1999.

[21] S. Kirkpatrick, Jr. C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, Vol. 220, pp. 671–680, 1983.

[22] N. E. Collins, R. W. Eglese, and B. L. Golden, "Simulated annealing—an annotated bibliography," American Journal of Mathematical and Management Sciences, Vol. 8, pp. 209–307, 1988.

[23] R. B. Rutenbar, "Simulated annealing algorithms: An overview," IEEE Circuits and Devices Magazine (January), pp. 19–26, 1989.

[24] R. W. Eglese, "Simulated annealing: A tool for operational research," European Journal of Operational Research Vol. 46, No. 3, pp. 271–281, 1990.

[25] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," Journal of the Operational Research Society, Vol. 57, No. 10, pp. 1143–1160, 2006.

[26] R. T. Clemen and T. Reilly, "Making hard decisions," Duxbury, Toronto, 2001.

[27] P. Andersen and N. C. Petersen, "A procedure for ranking efficient units in data envelopment analysis," Management Science, Vol. 39, pp. 1261–1264, 1993.

[28] D. Slottje, G. W. Scully, J. G. Hirschberg, and K. J. Hayes, "Measuring the quality of life across countries: A multidimensional analysis," Westview Press, Boulder, CO, 1991.

[29] M. Pinedo, "Scheduling theory: Algorithms and systems," Prentice-Hall, Inc., A Simon & Schuster Company Englewood Cliffs, New Jersey, pp. 10, 1995.

[30] A. Charnes, W. W. Cooper, and E. Rhodes, "Measuring the efficiency of decision making units," European Journal of Operational Research, Vol. 2, pp. 429–444, 1978.

