Journal of the Operational Research Society (2008) 59, 1109–1119 © 2008 Operational Research Society Ltd. All rights reserved. 0160-5682/08 $30.00
www.palgrave-journals.com/jors

A hybrid approach for single-machine tardiness problems with sequence-dependent setup times

S-W Lin[1]* and K-C Ying[2]

[1]Department of Information Management, Huafan University, Taipei, Taiwan; and [2]Department of Industrial Engineering and Management Information, Huafan University, Taipei, Taiwan

Scheduling problems in real systems often require sequence-dependent setup times. The topic of sequence-dependent setup times has not been addressed adequately, and improved competitiveness is thus not achieved. This study proposes a hybrid approach that takes advantage of simulated annealing (SA) and tabu search (TS) to solve single-machine tardiness problems with sequence-dependent setup times. To verify the proposed approach, experiments were conducted on benchmark problem sets that included both the weighted and un-weighted tardiness problems. The results show that the performance of the hybrid approach is superior to that of the SA, genetic algorithm, TS and ant colony optimization approaches, and is comparable with the Tabu-VNS approach. And the proposed approach found new upper bound values for many benchmark problems with an acceptable computational time.

Journal of the Operational Research Society (2008) **59,** 1109–1119. doi:10.1057/palgrave.jors.2602434
Published online 20 June 2007

Keywords: scheduling; hybrid approach; simulated annealing; tabu search; tardiness penalties; sequence dependence

1. Introduction

During the last 50 years, a lot of attention has been given to scheduling problems. One of the most common issues has been the single-machine scheduling (SMS) problem. The SMS problem does not necessarily involve only one machine. In linear flow operations, for example, such as an assembly group, where operations are sequentially performed on a batch of a product, a capacity constrained operation or bottleneck is frequently encountered. In such a case, the schedule of the bottleneck effectively schedules the entire operation. Another example is to treat a group of machines as a single unit. High-tech industrial manufacturing facilities, such as computer controlled machining centres and robotic cells, are often treated as an SMS problem for scheduling purposes.

Approximately three quarters of managers surveyed by Panwalkar *et al* (1973) stated that they scheduled at least one operation that required sequence-dependent setup times, while approximately 15% of managers reported that all operations required sequence-dependent setup times. Wortman (1992) has pointed out that sequence-dependent setup times have not been addressed adequately, and improved competitiveness is thus not achieved. Tan *et al* (2000) has pointed out that the assumption of sequence-independent setup times does not align with a significant number of real-world situations. Therefore, a lot of studies on this topic have offered heuristics or meta-heuristics for the scheduling problem with sequence-dependent setup times. An excellent review of scheduling research involving setup time considerations is provided for readers to explore the literature further (Allahverdi *et al*, 1999).

Meeting due dates was indicated by Panwalkar *et al* (1973) to be the most important scheduling criterion. Minimization of total weighted/un-weighted tardiness is frequently cited as a measure of performance. Although an easy to understand concept, tardiness is actually a difficult criterion to work with, even in the single-machine environment. There is no simple rule to minimize tardiness with sequence-independent setup times, except for two special cases: (i) the Shortest Processing Time, in which the schedule minimizes total tardiness if all jobs are tardy, and (ii) the Earliest Due Date, in which the schedule minimizes total tardiness if at most one job is tardy (Emmons, 1969). Considering the existing limitations, the objective of this study is to propose an efficient hybrid approach for solving the single-machine tardiness problems with sequence-dependent setup times.

The remainder of this paper is organized as follows. Section 2 presents a review of the literature on single-machine tardiness problems. Next, Section 3 elaborates on the proposed approach for solving single-machine total tardiness problems with sequence-dependent setup times. Section 4 compares the experimental results for benchmark problems, and conclusions and future research are drawn in Section 5.

**Correspondence: S-W Lin, Department of Information Management, Huafan University, No. 1 Huafan Road, Shihtin Hsiang, Taipei Hsien, Taiwan 223, Taiwan.*
E-mail: swlin@cc.hfu.edu.tw

2. Literature review

Single-machine total weighted tardiness problems with sequence-dependent setup times can be expressed as $1/s_{ij}/\sum w_iT_i$ and defined as follows. There are n jobs, each having an integer processing time p_j, a due date d_j, by which it should ideally be finished, a non-negative weight w_j, for the delay penalty, and a setup time s_{ij}, that is incurred when job j follows job i in the processing sequence. All jobs are available for processing at time zero, and will be processed without interruption on a continuously available machine that can handle only one job at a time. If the weight w_j is equal to 1 for all jobs, the problems belong to the single-machine total un-weighted tardiness problems with sequence-dependent setup times and can be expressed as $1/s_{ij}/\sum T_i$. Let Π be the processing sequence of the jobs, $\Pi = \{\Pi(0), \Pi(1), \ldots, \Pi(n)\}$, where $\Pi(i)$ is the index of the ith job in the sequence and $\Pi(0) = 0$. The completion time of the job in the ith position of the sequence can be calculated as $C_{\Pi(i)} = \sum_{k=1}^{i} [s_{\Pi(k-1)\Pi(k)} + p_{\Pi(k)}]$, the tardiness of the job in the ith position of the sequence is $T_{\Pi(i)} = \max[0, C_{\Pi(i)} - d_{\Pi(i)}]$, and the total weighted tardiness for a sequence Π is $WT_{\Pi} = \sum_{i=1}^{n} w_{\Pi(i)} T_{\Pi(i)}$. The objective is to determine the processing sequence of jobs so as to minimize the total weighted/un-weighted tardiness of jobs.

Lawler (1977) pointed out that the $1//\sum w_iT_i$ problem was strongly NP-hard. The proposed $1/s_{ij}/\sum w_iT_i$ problem is also strongly NP-hard because the incorporation of setup times complicates the problem. For such problems, there is a need to use heuristics to obtain a near optimal solution with an acceptable computational time.

Scheduling heuristics can be broadly classified into two categories: construction heuristics and improvement heuristics (Vepsalainen and Morton, 1987). Construction heuristics use dispatching rules to build a solution by fixing a job in a position at each step. To date, the best construction heuristic for the $1/s_{ij}/\sum w_iT_i$ problem has been the apparent tardiness cost with setups (ATCS) (Lee *et al*, 1997). Although ATCS can derive a feasible solution quickly, the solution quality is found inferior, especially for large problems.

In recent years, improvement heuristics have been extensively used to solve these problems. Rubin and Ragatz (1995) worked out a genetic search approach to solve $1/s_{ij}/\sum T_i$ problems. Tan and Narasimhan (1997) provided a simulated annealing (SA) approach to minimize the total un-weighted tardiness on a single-machine. Armentano and Mazzini (2000) proposed a genetic algorithm (GA) on a single-machine with setup times to minimize total un-weighted tardiness. Tan *et al* (2000) compared four methods, namely, branch-and-bound, genetic search, random-start pair-wise interchange, and SA for minimizing the total un-weighted tardiness on a single-machine with sequence-dependent setup times. Franca *et al* (2001) proposed a memetic algorithm for the total un-weighted tardiness SMS problem. Gagné *et al* (2002) proposed an ant colony optimizations (ACO) algorithm for the single-machine un-weighted tardiness problem with sequence-dependent setup times. Furthermore, Gagné *et al* (2005) proposed a hybrid Tabu-VNS meta-heuristic approach for solving the single-machine un-weighted tardiness problems with sequence-dependent setup times.

Cicirello and Smith (2005) provided 120 $1/s_{ij}/\sum w_iT_i$ problems and applied five stochastic sampling approaches, namely, limited discrepancy search, heuristic-biased stochastic sampling, value biased stochastic sampling (VBSS), value biased stochastic sampling seeded hill-climber (VBSS-HS), and SA approach to solve the problems they generated. Liao and Juan (2007) developed an ACO approach for single-machine tardiness scheduling with sequence-dependent setups and solved both the weighted and un-weighted tardiness problems provided by Cicirello and Smith (2005), Rubin and Ragatz (1995), and Gagné *et al* (2002). Their ACO approach proposed is basically the ACS version (Dorigo and Gambardella, 1997), and adopts several distinctive features that make it competitive with other algorithms. It introduces the feature of a minimum pheromone value from the 'max-min-ant system' (Stützle and Hoos, 2000), applies a new parameter for the initial pheromone trial, and adjusts the timing of local search. Lin and Ying (2007) proposed an SA approach, a GA, and a tabu search (TS) approach to solve the above 120 benchmark problems. In their study, a random swap and insertion search is applied to obtain the next solution in SA. The swap is done by randomly selecting the ith and jth number of current solution, and then swapping the values of these two numbers directly. The insertion is done by randomly selecting the ith number of the current solution, and inserting it into the position immediately preceding the jth number of the current solution. A mutation operator performed by a greedy local search is used in GA. The mutation starts by selecting one job and comparing swap and insertion options. The selected job is paired with all other jobs individually to identify the swap or insertion operation leading to the largest improvement of the objective function value. The operation that yields the largest improvement of the objective function is chosen and then performed. Similarly, a swap and an insertion tabu list are adopted in TS. In each iteration, randomly choose the ith number of the current solution, and the neighbourhood of the current solution is the set of all possible swaps and insertions of the ith number with the other numbers. They found new upper bound values for most benchmark problems within reasonable computational expense.

3. The proposed hybrid approach

This study proposes a hybrid approach that takes advantage of SA and TS for solving the single-machine tardiness scheduling problem. The proposed hybrid approach is an improvement on the SA approach developed by Lin and Ying (2007), which added the concept of TS to the SA approach. The main part of the proposed SA-Tabu approach is the SA, and the concept of TS is used to obtain the next solution.

The ideas and characteristics of SA and TS are described as follows.

Introduced by Metropolis *et al* (1953), and popularized by Kirkpatrick *et al* (1983), the concept of SA is taken from nature. Annealing is the process through which slow cooling of metal produces good and low energy state crystallization, whereas fast cooling produces poor crystallization. The optimization procedure of SA reaching a (near) global minimum mimics the crystallization cooling procedure. SA draws a random initial solution to begin its search. In each iteration, the algorithm takes a new solution from the predetermined neighbourhood of current solution. This new solution's objective function value is then compared to the current solution's objective function value in order to determine if the new solution's objective function value is better. For the case of minimization, if the new solution's objective function value is smaller, a new solution is automatically accepted and becomes the current solution from which the search will continue. The algorithm will then proceed with another step. A higher objective function value for the next solution may also be accepted as the current solution with a probability determined by the Metropolis criteria (Metropolis *et al*, 1953). The essential idea is not to restrict the search algorithm moves to those that decrease the objective function value, but also allow moves that can increase the objective function value. In principle, this allows a search algorithm to escape from a local minimum.

TS, initially suggested by Glover (1989, 1990) is an iterative improvement approach designed for obtaining (near) global optimum solutions to combinatorial optimization problems. The idea of TS can be described briefly as follows. Starting from an initial solution TS iteratively moves from the current solution X, to its most improved solution Y, in the neighbourhood of X, or if none exists, chooses the least worsening solution, until a superimposed stopping criterion becomes true. In order to avoid cycling, moves which would bring us back to a recently visited solution should be forbidden or declared tabu for a certain number of iterations. This is accomplished by keeping the attributes of the forbidden moves in a list, called a tabu list. The size of the tabu list must be large enough to prevent cycling, but small enough not to forbid too many moves. If a tabu move is superior to the best solution found so far by the search, then this move can be selected even though it is tabu, overriding the tabu restriction. This is called the aspiration criterion.

For the application of the SA-Tabu approach to the single-machine total tardiness problems with sequence-dependent setup times, the solution representation, the initial solution, the neighbourhood, the tabu move, the aspiration criteria, the parameters used, and procedure are discussed as follows.

3.1. Solution presentation and the initial solution

The solution presentation is the permutation of n numbers in the set $\{1, 2, \ldots, n\}$, where the ith number in the permutation denotes the job is the ith job to be processed. The initial solution is generated randomly.

3.2. Neighbourhood

The set $N(X)$ is the set of solutions neighbouring X. A neighbourhood is sampled either by insertion or swap at random. The insertion is carried out by randomly selecting the ith number of X and inserting it into the position immediately preceding the randomly selected jth number of X. The swap is performed by randomly selecting both the ith and jth number of X, and then swapping the values of these two numbers directly. A 50% probability exists of carrying out an insertion, and there is a 50% probability exists of carrying out a swap in obtaining the neighbourhood solution of X. Given the current solution X, there are $n \times (n-1)$ possible insertion solutions and $n \times (n-1)/2$ possible swap solutions of X. However, only one of them is selected to evaluate the objective function value and to determine whether it is a tabu solution or not in one iteration.

3.3. Tabu move and aspiration criterion

In order to avoiding cycling, the tabu_time matrix is applied to impose tabu moves. Let (i, j) element of the tabu_time matrix contain the iteration number at which job i is allowed to return to the jth position of the scheduling sequence. The duration which a job is not allowed to move to a position of the scheduling sequence is called the tenure of the tabu move. If the chosen neighbourhood solution Y, of the current solution X, belongs to the tabu moves, then solution Y is discarded and the new neighbourhood is generated until Y does not belong to the tabu moves, or Y is the best solution found so far by the search.

3.4. Parameters used

The SA-Tabu begins with seven parameters, namely I_{iter},T_0, T_F, α, $S_{\max}$, MIN_T and MAX_T, where I_{iter} denotes the number of iterations the search proceeds at a particular temperature, T_0 represents the initial temperature, T_F represents the final temperature that stops the SA-Tabu procedure if the current temperature is lower than T_F. α is the coefficient controlling the cooling schedule, $S_{\max}$ is the maximum number of total evaluations (possible schedules), MIN_T and MAX_T are the minimal and maximal tenure of tabu moves, respectively.

3.5. The SA-Tabu procedure

In the beginning the current temperature T is set to be the same as T_0. Next, an initial solution X is randomly generated. The current best solution X_{best} is set to be equal to X, the current objective function value F_{cur} is set to be equal to the objective function value of X, F_X, and the best objective function value obtained so far F_{best} is set to be equal to F_{cur}.

In each iteration, the next solution Y is generated from $N(X)$ and its objective function value is evaluated. The new solution Y cannot belong to the set of tabu solutions, unless it is the best solution found so far in the procedure. T is decreased after running I_{iter} iterations from the previous

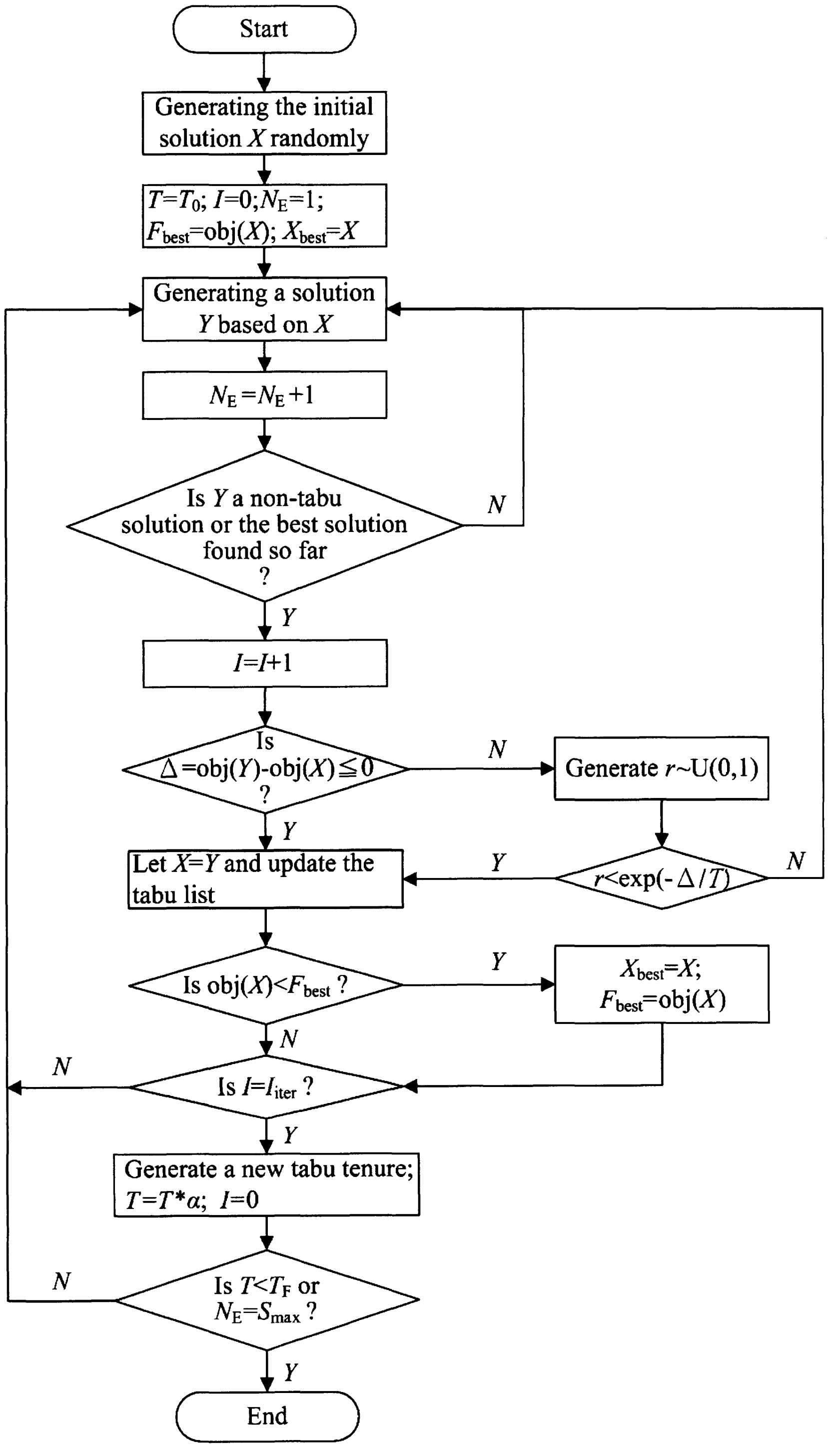

Figure 1 The flowchart of the proposed SA-Tabu approach.

decrease, according to a formula $T := \alpha T$, where $0 < \alpha < 1$. The tenure of a tabu move is re-assigned by choosing an integral value between MIN_T and MAX_T randomly, when T is decreased once.

Let obj(X) denote the calculation of the objective function value of X, and Δ denote the difference between obj(X) and obj(Y); that is $\Delta = \text{obj}(Y) - \text{obj}(X)$. The probability of replacing X with Y, where X is the current solution and Y is the next solution, given that $\Delta > 0$, is $\exp(-\Delta/T)$, is accomplished by generating a random number $r \in [0, 1]$ and replacing the solution X with Y if $r < \exp(-\Delta/T)$. Meanwhile, if $\Delta < 0$, the probability of replacing X with Y is 1. If the solution X is replaced by Y, the tabu_time matrix is changed accordingly. If T is lower than T_F, or the number of total evaluations (possible schedules) N_E reaches a pre-specified number S_{max}, the algorithm is terminated. The X_{best} records the best solution as the algorithm progresses. Following the termination of SA-Tabu procedure, the (near) global optimal schedule can thus be derived by X_{best}. The flowchart of the proposed SA-Tabu approach can be seen in Figure 1.

4. Experimental results

The proposed approach was implemented in C language and run on a P-IV 1.4 GHz PC with 512 MB RAM. Experiments were carried out to compare the proposed approach with benchmark approaches (Cicirello and Smith, 2005; Liao and Juan, 2007; Lin and Ying, 2007) for total weighted tardiness SMS problems with sequence-dependent setup times, and benchmark approaches (Rubin and Ragatz, 1995; Gagné *et al*, 2002, 2005; Liao and Juan, 2007) for total un-weighted tardiness SMS problems with sequence-dependent setup times.

For the total weighted tardiness SMS problems ($1/s_{ij}/\sum w_i T_i$), the benchmark problem instances provided by Cicirello and Smith (2005) were generated according to a procedure described by Lee *et al* (1997). Each problem instance is characterized by three parameters: the due-date tightness factor τ; the due-date range factor R; and the setup time severity factor η. Consider problem sets characterized by the following parameter values: $\tau = \{0.3, 0.6, 0.9\}$; $R = \{0.25, 0.75\}$; $\eta = \{0.25, 0.75\}$. Each of the 12 combinations of parameters had 10 problem instances with the number of jobs n, equal to 60. These 12 problem sets cover a spectrum from loosely to tightly constrained problem instances. The benchmark problems can be obtained via http://www.ozone.ri.cmu.edu/benchmarks.html.

In order to equitably compare the proposed SA-Tabu approach with the old approaches, the maximum number of total evaluations (possible schedules) S_{max} of the proposed approach was set at 20 000 000 which is the same as the maximum number of total evaluations used in the old SA approach (Cicirello and Smith, 2005; Lin and Ying, 2007), GA and TS approaches (Lin and Ying, 2007). After running a few problems with several combinations of parameters, the parameter values for SA-Tabu were $I_{iter} = n \times 2, 400$, $T_0 = 100$, $T_F = 1$, $\alpha = 0.965$, $MIN_T = 3$, and $MAX_T = 6$, where n is the number of jobs to be scheduled. Each problem was solved 10 times for the proposed approach. Since the maximum number of total evaluations (possible schedules) is the same for all problems, the average running time of SA-Tabu was almost equal and was about 27 s per trial for each problem. The benchmark value (Cicirello and Smith, 2005) and the best objective function value for each problem instance obtained by the ACO approach (denoted as ACO_{LJ}) (Liao and Juan, 2007), SA, GA and TS (denoted as SA_{LY}, GA_{LY} and TS_{LY}) (Lin and Ying, 2007) are shown in Table 1. The best objective function value for each instance obtained by ACO, SA, GA and TS approaches are cited from their original papers. A plus sign or a minus sign is added in the back of the obtained objective function value to indicate that the obtained value is better or worse than the benchmark value. If the objective function value obtained was the best among five approaches (ACO_{LJ}, SA_{LY}, GA_{LY}, TS_{LY} and SA-Tabu) and was better than the benchmark value, the objective function value is marked with asterisk sign. Details of the solutions obtained in this research are displayed online at http://pc33.mis.hfu.edu.tw/smswtsds/.

It can be observed from Table 1, that all five approaches find optimal solutions for those 16 instances whose optimal solutions are known (solutions with zero tardiness). For the remaining 104 instances, ACO_{LJ} is the best for 12 (12/104 = 11.54%) instances, SA_{LY} is the best for 22 (22/104 = 21.15%), GA_{LY} is the best for 25 (25/104 = 24.04%), TS_{LY} is the best for (21/104 = 20.19%), and SA-Tabu is the best for 31 (31/104 = 29.81%) such instances, respectively; five instances end in a tie. Therefore, the proposed SA-Tabu approach obtains a higher number of solutions that are the optimal solutions, or the best solutions found so far in the literature.

To calculate the average improvement rates of other approaches and the proposed SA-Tabu approach, the formulas $(f^{bm} - f^{ACO})/f^{bm} \times 100\%$, $(f^{bm} - f^{SA})/f^{bm} \times 100\%$, $(f^{bm} - f^{GA})/f^{bm} \times 100\%$, $(f^{bm} - f^{TS})/f^{bm} \times 100\%$, and $(f^{bm} - f^{SA\text{-}Tabu})/f^{bm} \times 100\%$ are used to calculate the improvement rate for each problem of ACO_{LJ}, SA_{LY}, GA_{LY}, TS_{LY}, and SA-Tabu, respectively. f^{ACO}, f^{SA}, f^{GA}, f^{TS}, and $f^{SA\text{-}Tabu}$ denote the best objective function values from 10 trials (executed for each individual problem 10 times) of the ACO_{LJ}, SA_{LY}, GA_{LY}, TS_{LY}, and SA-Tabu approaches, respectively, where f^{bm} denotes the objective function value of the benchmark value (Cicirello and Smith, 2005). The average improvement rates of the ACO_{LJ}, SA_{LY}, GA_{LY}, TS_{LY}, and SA-Tabu approaches for each of the 12 problem sets are shown in Table 2. Comparing the positive average improvement rates, the proposed SA-Tabu approach has superior results for the five problem sets 2, 6, 7, 8, and 9. The average improvement rates of 120 problems for the ACO_{LJ}, SA_{LY}, GA_{LY}, TS_{LY}, and SA-Tabu approaches were 4.10, 9.32, 9.97, 8.90, and 10.22%, respectively. In the context of 120 problems, the proposed SA-Tabu approach has the best average improvement rate among the five approaches.

To verify whether the proposed SA-Tabu approach is better than other approaches, a paired t-test on the best objective function value obtained is used to compare the proposed SA-Tabu approach with ACO, SA, GA, and TS. When the confidence level α is set to be 0.05, there is significant difference between the proposed SA-Tabu approach and other approaches. That is to say, the proposed SA-Tabu approach outperforms the ACO, SA, GA and TS approaches. The statistical results are shown in Table 3.

Because the maximum number of total evaluations (possible schedules) of the proposed approach were set at

Table 1 Comparison between benchmark values and five approaches for weighted tardiness problems

Prob.	*Benchmark*	ACO_{LJ}	SA_{LY}	GA_{LY}	TS_{LY}	*SA-Tabu*
001	978	894+	732+	684*	738+	720+
002	6489	6307+	5487+	5142+	5082*	5367+
003	2348	2003+	1792*	1798+	1798+	1805+
004	8311	8003+	6644+	6526*	6726+	6646+
005	5606	5215+	4919+	4662*	4891+	4828+
006	8244	5788*	7213+	7348+	7728+	7558+
007	4347	4150+	3837+	3693*	3799+	3783+
008	327	159+	166+	142*	171+	171+
009	7598	7490+	6653+	6349+	6729+	6264*
010	2451	2345+	2021*	2038+	2165+	2119+
011	5263	5093+	4660+	3867*	4938+	4360+
012	0	0	0	0	0	0
013	6147	5962+	6223-	5685*	5896+	5719+
014	3941	4035+	3233+	3045*	3351+	3357+
015	2915	2823+	1885+	1458*	1843+	1727+
016	6711	6153+	5614+	4940*	5640+	5461+
017	462	443+	204*	221+	258+	213+
018	2514	2059+	1698+	1610*	1945+	1650+
019	279	265+	228+	208+	231+	36*
020	4193	4204-	3250+	2967*	3243+	3543+
021	0	0	0	0	0	0
022	0	0	0	0	0	0
023	0	0	0	0	0	0
024	1791	1551+	1098+	1063*	1100+	1103+
025	0	0	0	0	0	0
026	0	0	0	0	0	0
027	229	137+	0*	0*	0*	0*
028	72	19+	0*	0*	0*	0*
029	0	0	0	0	0	0
030	575	372+	165*	180+	269+	201+
031	0	0	0	0	0	0
032	0	0	0	0	0	0
033	0	0	0	0	0	0
034	0	0	0	0	0	0
035	0	0	0	0	0	0
036	0	0	0	0	0	0
037	2407	2078+	755*	862+	1009+	970+
038	0	0	0	0	0	0
039	0	0	0	0	0	0
040	0	0	0	0	0	0
041	73176	73578-	71593+	71855+	71186*	71193+
042	61859	60914+	58802+	59515+	58199*	58779+
043	149990	149670+	147211*	148283+	147307+	148211+
044	38726	37390+	35648*	35765+	35870+	35789+
045	62760	62535+	59629+	59307*	59828+	59487+
046	37992	38779-	35948+	35698+	35320*	36019+
047	77189	76011+	73984+	74997+	74277+	73972*
048	68920	68852+	65164*	65586+	65813+	65440+
049	84143	81530+	79331+	79635+	79055*	79172+
050	36235	35507+	32809+	33081+	32797+	32743*
051	58574	55794+	53449+	52639+	54292+	52163*
052	105367	105203+	100090+	99200+	101756+	98316*
053	95452	96218-	91302*	91794+	95108+	92924+
054	123558	124132-	128915-	129437-	127321-	122968*
055	76368	74469+	72051+	69776+	72819+	69571*
056	88420	87474+	81324+	78960*	80280+	79024+
057	70414	67447+	69320*	71030-	71522-	70974-
058	55522	52752+	48976+	49168+	48081*	49876+
059	59060	56902+	55396*	55878+	56190+	56219+
060	73328	72600+	69000+	68851+	69104+	68176*

Table 1 continued.

Prob.	*Benchmark*	ACO_{LJ}	SA_{LY}	GA_{LY}	TS_{LY}	*SA-Tabu*
061	79884	80343-	76396+	76888+	76454+	76357*
062	47860	46466+	45162+	44976+	44769*	44769*
063	78822	78081+	75317*	76098+	76591+	75951+
064	96378	95113+	92572*	93376+	92698+	92591+
065	134881	132078+	127912+	129243+	128765+	126907*
066	64054	63278+	59832+	61316+	60078+	59717*
067	34899	32315+	29394+	29390*	29394+	29390*
068	26404	26366+	22516+	22861+	22148*	22199+
069	75414	64632*	71652+	72309+	71287+	71148+
070	81200	81356-	75644+	75102*	75552+	75102*
071	161233	156272+	152890+	150709+	153181+	150053*
072	56934	54849+	49669+	48735+	46903*	47288+
073	36465	34082+	29408*	30289+	29488+	30017+
074	38292	33725+	33525+	35982+	33375*	34578+
075	30980	27248+	22912+	22591+	21863*	23851+
076	67553	66847+	55055*	58307+	58763+	55619+
077	40558	37257+	34732+	34929+	35038+	33303*
078	25105	24795+	21697+	22154+	21493+	20632*
079	125824	122051+	123544+	121118+	122336+	119099*
080	31844	26470+	22344+	20335+	20571+	20161*
081	387148	387866-	385471+	384996*	386303+	386046+
082	413448	413181+	410979+	412684+	410996+	410458*
083	466070	464443+	461242+	462701+	460978+	459817*
084	331659	330714+	330384*	331315+	331470+	330762+
085	558556	562083-	555106*	555433+	556810+	555922+
086	365783	365199+	364381+	364931+	365055+	364075*
087	403016	401535+	399876+	399439+	400697+	399214*
088	436855	436925-	436343+	434948*	436253+	435217+
089	416916	412359+	410966*	414748+	410993+	411090+
090	406939	404105+	403702+	402233*	402856+	402647+
091	347175	345421+	346267+	346616+	344988*	346978+
092	365779	365217+	365699+	365129+	365824-	364593*
093	410462	412986-	416965-	415950-	416350-	413091-
094	336299	335550+	341466-	343237-	339381-	335210*
095	527909	526916+	532185-	521512+	527078-	519364*
096	464403	461484*	467161-	472178-	464121+	469262-
097	420287	419370+	419269+	419805+	413109*	418428+
098	532519	533106-	534518-	538715-	533242-	535902-
099	374781	370084*	375263-	381134-	374847-	376488-
100	441888	441794+	442627-	439944*	446324-	444462-
101	355822	355372+	353959+	354158+	353408*	353582+
102	496131	495980+	496304-	495877+	493889*	495675+
103	380170	379913*	380790-	381471-	380706-	380370-
104	362008	360756+	358334+	359641+	358222*	359246+
105	456364	454890+	450808*	455037+	452249+	452415+
106	459925	459615+	455927+	455849*	457167+	456580+
107	356645	354097+	353371+	357084-	354312+	352855*
108	468111	466063+	463416+	464797+	462737*	464196+
109	415817	414896+	413205*	415710+	414095+	414221+
110	421282	421060+	421039+	422035-	419481*	419749+
111	350723	347233*	351289-	350062+	349195+	348296+
112	377418	373238*	373548+	378602-	377212+	377158+
113	263200	262367+	263048+	261239*	261737+	261554+
114	473197	470327*	473679-	473408-	472617+	470963+
115	460225	459194*	468101-	464158-	470431-	470264-
116	540231	527459*	546630-	542450-	542873-	538553+
117	518579	512286+	518084+	525999+	513125+	512028*
118	357575	352118*	356591+	360883-	358349-	356291+
119	583947	584052-	581553+	579708+	579462*	581740+
120	399700	398590*	403834-	407996-	406913-	402844-

Table 2 Average improvement rates (%) of the SA-Tabu approach against other approaches

Set no	*Problem set* ($n = 60$)	ACO_{LJ}	SA_{LY}	GA_{LY}	TS_{LY}	*SA-Tabu*
1	Problem 001 to 010 ($\tau = 0.3$, $R = 0.25$, $\eta = 0.25$)	12.82	20.00	22.83	19.12	20.08
2	Problem 011 to 020 ($\tau = 0.3$, $R = 0.25$, $\eta = 0.75$)	4.23	20.89	27.60	18.46	28.92
3	Problem 021 to 030 ($\tau = 0.3$, $R = 0.75$, $\eta = 0.25$)	16.25	30.39	30.93	29.18	30.35
4	Problem 031 to 040 ($\tau = 0.3$, $R = 0.75$, $\eta = 0.75$)	1.37	6.86	6.42	5.81	5.97
5	Problem 041 to 050 ($\tau = 0.6$, $R = 0.25$, $\eta = 0.25$)	0.97	5.21	4.77	5.33	5.16
6	Problem 051 to 060 ($\tau = 0.6$, $R = 0.25$, $\eta = 0.75$)	2.10	5.29	5.65	4.44	6.15
7	Problem 061 to 070 ($\tau = 0.6$, $R = 0.75$, $\eta = 0.25$)	2.95	7.25	6.56	7.25	7.60
8	Problem 071 to 080 ($\tau = 0.6$, $R = 0.75$, $\eta = 0.75$)	6.75	15.39	15.02	16.32	16.97
9	Problem 081 to 090 ($\tau = 0.9$, $R = 0.25$, $\eta = 0.25$)	0.45	0.66	0.56	0.56	0.73
10	Problem 091 to 100 ($\tau = 0.9$, $R = 0.25$, $\eta = 0.75$)	0.25	−0.47	−0.50	−0.11	−0.06
11	Problem 101 to 110 ($\tau = 0.9$, $R = 0.75$, $\eta = 0.25$)	0.24	0.60	0.24	0.64	0.57
12	Problem 111 to 120 ($\tau = 0.9$, $R = 0.75$, $\eta = 0.75$)	0.86	−0.23	−0.44	−0.23	0.12
	Average improvement rate of 120 problems	4.10	9.32	9.97	8.90	10.22

Table 3 Paired t-test on the best objective function obtained of weighted tardiness problems between the proposed SA-Tabu approach against other approaches

H_0 H_1	$SA\text{-}Tabu = ACO_{LJ}$ $SA\text{-}Tabu \neq ACO_{LJ}$	$SA\text{-}Tabu = SA_{LY}$ $SA\text{-}Tabu \neq SA_{LY}$	$SA\text{-}Tabu = GA_{LY}$ $SA\text{-}Tabu \neq GA_{LY}$	$SA\text{-}Tabu = TS_{LY}$ $SA\text{-}Tabu \neq TS_{LY}$
Paired difference (mean)	−1040.717	−492.667	−721.467	−414.825
t-value	−3.446	−2.732	−3.719	−2.714
Degree of freedom	119	119	119	119
Two-tail significant	0.000	0.007	0.000	0.008

20 000 000, which is the same as the maximum number of total evaluations used in the SA approach (Cicirello and Smith, 2005; Lin and Ying, 2007), and used in GA and TS approaches (Lin and Ying, 2007), the computational time of SA_{LY}, GA_{LY}, TS_{LY}, and SA-Tabu is almost the same. However, judged by the number of best solutions found and the average improvement rate, the SA-Tabu approach is better than SA_{LY}, GA_{LY} and TS_{LY} approach. Although the computational time of ACO_{LJ} is much shorter than that of the proposed SA-Tabu approach, the number of best solutions found so far and the average improvement rate of ACO_{LJ} is worse than that of the proposed SA-Tabu approach.

For the un-weighted tardiness SMS problems $(1/s_{ij}/\sum T_i)$, we compared the SA-Tabu approach with the four best-performing approaches, RSPI, ACO_{GPJ}, ACO_{LJ}, and Tabu-VNS. The RSPI approach is a GA proposed by Rubin and Ragatz (1995), while the ACO_{GPG} approach is an ACO algorithm developed by Gagné *et al* (2002), and the Tabu-VNS approach is a hybrid TS/variable neighbourhood search algorithm developed by Gagné *et al* (2005). ACO_{LJ} is an ACO approach developed by Liao and Juan (2007). To make our SA-Tabu approach more efficient and effective, the parameter values for SA-Tabu were $I_{iter} = n \times 2,400$, $T_0 = 100$, $T_F = 1$, $\alpha = 0.965$, $MIN_T = 3$, and $MAX_T = 6$, where n is the number of jobs to be scheduled. The value of S_{max} is not set for the un-weighted tardiness SMS problems.

To test the small-and medium-sized problems (with 15–45 jobs), we used the 32 benchmark problem instances provided by Rubin and Ragatz (1995), which can be obtained at http://www.msu.edu/~rubin/research.html. Table 4 shows the comparison among RSPI, Tabu-VNS, ACO_{GPG}, ACO_{LJ}, and SA-Tabu. RSPI, ACO_{GPG}, and SA-Tabu gave the best solution from 20 trials, while Tabu-VNS and ACO_{LJ} gave the best solution from 10 trials. All five approaches find the optimal solution for those 16 instances whose optimal solutions are known. For the remaining 16 instances, the five approaches (RSPI, Tabu-VNS, ACO_{GPG}, ACO_{LJ}, and SA-Tabu) obtained the best solution for 3 (3/16 = 18.75%), 11 (11/16 = 68.75%), 2 (2/16 = 12.50%), 5 (5/16 = 31.25%), and 10 (10/16 = 62.50%) such instances respectively; seven instances end in a tie.

To test the large-sized problems (with 55–85 jobs), we used the benchmark problem instances provided by Gagné *et al* (2002) and made the comparison among Tabu-VNS, ACO_{GPG}, ACO_{LJ}, and SA-Tabu. The Tabu-VNS gave the best solutions from 10 trials, while the ACO_{GPG}, ACO_{LJ}, and SA-Tabu gave their best solutions from 20 trials. It can be observed from Table 5, that all four approaches found optimal solutions for those 10 instances whose optimal solutions are known (solutions with zero tardiness). For the remaining 22 instances, Tabu-VNS is the best for 9 (9/22 = 40.91%) instances, ACO_{LJ} is the best for 7 (7/22 = 31.82%), and SA-Tabu is the best for 8 (8/22 = 36.36%). Two instances

Table 4 Comparison of the proposed SA-Tabu approach with four best-performing approaches for the un-weighted tardiness problems (small-and medium-sized problems)

Problem	*No. of jobs*	*RSPI*	*Tabu-VNS*	*ACO_{GPG}*	*ACO_{LJ}*	*SA-Tabu*
Prob401	15	90*	90*	90*	90*	90*
Prob402	15	0*	0*	0*	0*	0*
Prob403	15	3418*	3418*	3418*	3418*	3418*
Prob404	15	1067*	1067*	1067*	1067*	1067*
Prob405	15	0*	0*	0*	0*	0*
Prob406	15	0*	0*	0*	0*	0*
Prob407	15	1861*	1861*	1861*	1861*	1861*
Prob408	15	5660*	5660*	5660*	5660*	5660*
Prob501	25	266	261+	261+	263	261+
Prob502	25	0*	0*	0*	0*	0*
Prob503	25	3497+	3503	3497+	3497+	3497+
Prob504	25	0*	0*	0*	0*	0*
Prob505	25	0*	0*	0*	0*	0*
Prob506	25	0*	0*	0*	0*	0*
Prob507	25	7225+	7225+	7268	7225+	7225+
Prob508	25	1915+	1915+	1945	1915+	1915+
Prob601	35	36	12+	16	14	17
Prob602	35	0*	0*	0*	0*	0*
Prob603	35	17792	17605	17685	17654	17587+
Prob604	35	19238	19168	19213	19092+	19092+
Prob605	35	273	228+	247	240	232
Prob606	35	0*	0*	0*	0*	0*
Prob607	35	13048	12969+	13088	13010	12969+
Prob608	35	4733	4732+	4733	4732+	4732+
Prob701	45	118	98+	103	103	107
Prob702	45	0*	0*	0*	0*	0*
Prob703	45	26745	26506+	26663	26568	26532
Prob704	45	15415	15213	15495	15409	15206+
Prob705	45	254	200+	222	219	216
Prob706	45	0*	0*	0*	0*	0*
Prob707	45	24218	23804+	24017	23931	23814
Prob708	45	23158	22873	23351	23028	22807+

*Indicate the optimal solution.
+The best performance among the five approaches (ties for all are not indicated).

end in a tie. The average computational time of each run for ACO_{LJ} is 24.17 s for all 32 instances on a P-IV 2.8 GHz PC, compared with the average computational time of each run for SA-Tabu, which is 39.73 s for all the 32 instances on a P-IV 1.4 GHz PC. The computational time of Tabu-VNS is not reported in the literature.

To verify whether the proposed SA-Tabu approach is better than other approaches, a paired t-test on the best objective function value obtained is used to compare the proposed SA-Tabu approach with the RSPI, Tabu-VNS, ACO_{GPG}, and ACO_{LJ} approaches. The statistical results are shown in Tables 6 and 7 for small-and medium sized problems and large-sized problems, respectively.

For the small-and medium-sized problems with un-weighted tardiness, when the confidence level α is set to be 0.05, there is significant evidence that the proposed SA-Tabu approach is better than the RSPI, ACO_{GPG}, and ACO_{LJ} approaches. Because there is no significant difference between SA-Tabu and Tabu-VNS, the effectiveness of the proposed SA-Tabu approach is the same as that of Tabu-VNS.

For the larger-sized problems with un-weighted tardiness, if the confidence level α is also set to be 0.05, there is significant evidence that the proposed SA-Tabu approach is better than ACO_{GPG} approach. However, the proposed SA-Tabu approach does not exhibit a significant difference between the Tabu-VNS and ACO_{LJ} approaches. That is to say, the effectiveness of the proposed SA-Tabu approach is the same as that of Tabu-VNS and ACO_{LJ}.

Nevertheless, the effectiveness of the proposed SA-Tabu approach is superior to that of the ACO_{LJ} approach on the weighted tardiness problems and un-weighted tardiness problems with small-and medium-sized scale. On the other hand, because the Tabu-VNS did not solve the weighted tardiness problems, further comparisons cannot be made. From the

Table 5 Comparison of the proposed SA-Tabu approach with three best-performing approaches for the un-weighted tardiness problems (large-sized problems)

Problem	*No. of jobs*	*Tabu-VNS*	ACO_{GPG}	ACO_{LJ}	*SA-Tabu*
Prob551	55	185	212	183^+	230
Prob552	55	0^*	0^*	0^*	0^*
Prob553	55	40644	40828	40676	40607^+
Prob554	55	14711	15091	14684	14653^+
Prob555	55	0^*	0^*	0^*	0^*
Prob556	55	0^*	0^*	0^*	0^*
Prob557	55	35841	36489	36420	35835^+
Prob558	55	19872	20624	19888	19871^+
Prob651	65	268^+	295	268^+	316
Prob652	65	0^*	0^*	0^*	0^*
Prob653	65	57602	57779	57584^+	57762
Prob654	65	34466	34468	34306^+	34378
Prob655	65	2^+	13	7	25
Prob656	65	0^*	0^*	0^*	0^*
Prob657	65	55080^+	56246	55389	55196
Prob658	65	27187	29308	27208	27114^+
Prob751	75	241^+	263	241^+	299
Prob752	75	0^*	0^*	0^*	0^*
Prob753	75	77739	78211	77663^+	78151
Prob754	75	35709	35826	35630	35250^+
Prob755	75	0^*	0^*	0^*	0^*
Prob756	75	0^*	0^*	0^*	0^*
Prob757	75	59763^+	61513	60108	60144
Prob758	75	38789	40277	38704	38431^+
Prob851	85	384^+	453	455	447
Prob852	85	0^*	0^*	0^*	0^*
Prob853	85	97880^+	98540	98443	98187
Prob854	85	80122	80693	79553^+	80037
Prob855	85	283^+	333	324	354
Prob856	85	0^*	0^*	0^*	0^*
Prob857	85	87244^+	89654	87504	87900
Prob858	85	75533	77919	75506	75029^+

*Indicate the optimal solution.
+The best performance among the four approaches (ties for all are not indicated).

Table 6 Paired t-test on the best objective function obtained of un-weighted tardiness problems (small-and medium-sized) between SA-Tabu and other approaches

H_0 H_1	*SA-Tabu = RSPI* *SA-Tabu ≠ RSPI*	*SA-Tabu = Tabu-VNS* *SA-Tabu ≠ Tabu-VNS*	*SA-Tabu* = ACO_{GPG} *SA-Tabu* ≠ ACO_{GPG}	*SA-Tabu* = ACO_{LJ} *SA-Tabu* ≠ ACO_{LJ}
Paired difference (mean)	−53.813	−3.219	−49.844	−21.594
t-value	−2.841	−0.951	−2.483	−2.191
Degree of freedom	31	31	31	31
Two-tail significant	0.008	0.349	0.019	0.036

Table 7 Paired t-test on the best objective function obtained of un-weighted tardiness problems (large-sized) between SA-Tabu and other approaches

H_0 H_1	*SA-Tabu = Tabu-VNS* *SA-Tabu ≠ Tabu-VNS*	*SA-Tabu* = ACO_{GPG} *SA-Tabu* ≠ ACO_{GPG}	*SA-Tabu* = ACO_{LJ} *SA-Tabu* ≠ ACO_{LJ}
Paired difference (mean)	20.969	−463.094	−16.500
t-value	0.539	−3.447	−0.414
Degree of freedom	31	31	31
Two-tail significant	0.297	0.002	0.297

viewpoint of a better solution being obtained, an important contribution of the study is to provide new objective function upper bounds for an existing set of benchmark problems.

5. Conclusions and future research

To solve single-machine total weighted/un-weighted tardiness problems with sequence-dependent setup times, this study proposed a hybrid SA and TS approach. The computational results demonstrate that the proposed SA-Tabu approach performs well by identifying lower total penalties, and thus this approach identifies new upper bounds for many benchmark problems. Therefore, the experimental results indicate that the proposed SA-Tabu approach can be efficiently applied to SMS problems with weighted/un-weighted tardiness penalties and setup time considerations.

Possible directions for future research include; employing other meta-heuristics, such as the hybrid GA and the TS approach, and extending the proposed approach to the multi-machine tardiness problems with sequence-dependent setup times.

Acknowledgements—We are grateful to the anonymous referees for their valuable suggestions and comments.

References

Allahverdi A, Gupta JND and Aldowaisan T (1999). A review of scheduling research involving setup considerations. *Omega* **27**: 219–239.

Armentano VA and Mazzini R (2000). A genetic algorithm for scheduling on a single machine with set-up times and due dates. *Prod Plan Control* **11**: 713–720.

Cicirello VA and Smith SF (2005). Enhancing stochastic search performance by value-based randomization of heuristics. *J Heuristics* **11**: 5–34.

Dorigo M and Gambardella LM (1997). Ant colony system: A cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* **1**: 53–66.

Emmons H (1969). One machine sequencing to minimize certain functions of job tardiness. *Opns Res* **17**: 701–715.

Franca PM, Mendes A and Moscato P (2001). A memetic algorithm for the total tardiness single machine scheduling problem. *Eur J Opl Res* **132**: 224–242.

Gagné C, Gravel M and Price WL (2005). Using metaheuristic compromise programming for the solution of multiple-objective scheduling problems. *J Opl Res Soc* **56**: 687–698.

Gagné C, Price WL and Gravel M (2002). Comparing an ACO algorithm with other heuristics for the single machine scheduling problem with sequence-dependent setup times. *J Opl Res Soc* **53**: 895–906.

Glover F (1989). Tabu search—Part I. *ORSA J Comput* **1**: 190–206.

Glover F (1990). Tabu search—Part II. *ORSA J Comput* **2**: 4–32.

Kirkpatrick S, Gelatt Jr. CD and Vecchi MP (1983). Optimization by simulated annealing. *Science* **220**: 671–680.

Lawler EL (1977). A 'pseudopolynomial' algorithm for sequencing jobs to minimize total tardiness. *Ann Disc Math* **1**: 331–342.

Lee YH, Bhaskaran K and Pinedo M (1997). A heuristic to minimize the total weighted tardiness with sequence-dependent setups. *IIE Trans* **29**: 45–52.

Liao C-J and Juan H-C (2007). An ant colony optimization for single-machine tardiness scheduling with sequence-dependent setups. *Comput Opl Res* **34**: 1899–1909.

Lin S-W and Ying K-C (2007). Solving single machine total weighted tardiness problems with sequence-dependent setup times by meta-heuristics. *Int J Adv Manuf Tech* (in press, DOI: 10.1007/s00170-006-0693-1).

Metropolis N, Rosenbluth AW, Rosenbluth MN, Teller AH and Teller E (1953). Equation of state calculations by fast computing machines. *J Chem Phys* **21**: 1087–1092.

Panwalkar SS, Dudek RA and Smith ML (1973). Sequencing research and the industrial scheduling problem. In: Elmaghraby SE (ed). *Symposium on the Theory of Scheduling and its Applications*. Springer-Verlag: Berlin, pp 29–38.

Rubin PA and Ragatz GL (1995). Scheduling in a sequence dependent setup environment with genetic search. *Comput Opl Res* **22**: 85–99.

Stützle T and Hoos HH (2000). Max–min ant system. *Future Gen Comput Syst* **16**: 889–914.

Tan KC and Narasimhan R (1997). Minimizing tardiness on a single processor with sequence-dependent setup times: A simulated annealing approach. *Omega* **25**: 619–634.

Tan KC, Narasimhan R, Rubin PA and Ragatz GL (2000). A comparison of four methods for minimizing total tardiness on a single processor with sequence dependent setup times. *Omega* **28**: 313–326.

Vepsalainen APJ and Morton TE (1987). Priority rules for job shops with weighted tardiness cost. *Mngt Sci* **33**: 1035–1047.

Wortman DB (1992). Managing capacity: Getting the most from your company's assets. *Indust Eng* **24**: 47–49.

Received July 2006;
accepted March 2007 after three revisions