Journal of Scheduling **8:** 355–386, 2005.

ORDER SCHEDULING IN AN ENVIRONMENT WITH DEDICATED RESOURCES IN PARALLEL

JOSEPH Y.-T. LEUNG[1], HAIBING LI[1] AND MICHAEL PINEDO[2]

[1]*Department of Computer Science, New Jersey Institute of Technology, Newark, NJ 07102, USA*
[2]*Stern School of Business, New York University, 40 West Fourth Street, New York, NY 10012, USA*

ABSTRACT

We consider m machines in parallel with each machine capable of producing one specific product type. There are n orders with each one requesting specific quantities of the various different product types. Order j may have a release date r_j and a due date d_j. The different product types for order j can be produced at the same time. We consider the class of objectives $\sum f_j(C_j)$ that includes objectives such as the total weighted completion time $\sum w_j C_j$ and the total weighted tardiness $\sum w_j T_j$ of the n orders. We present structural properties of the various problems and a complexity result. In particular, we show that minimizing $\sum C_j$ when $m \geq 3$ is strongly NP-hard. We introduce two new heuristics for the $\sum C_j$ objective. An empirical analysis shows that our heuristics outperform all heuristics that have been proposed for this problem in the literature.

KEY WORDS: order scheduling, total completion time, NP-hard, approximation, Tabu Search

1. INTRODUCTION

Consider a facility with m different machines in parallel. Each machine can produce one particular product type. To produce one unit of product type i on machine i requires a processing time t_i. Assume there are n orders that come from n different clients. Order j requests a quantity of product type i, $i = 1, \ldots, m$, $j = 1, \ldots, n$. The total processing time needed to produce the items of type i for order j is p_{ij}. Order j has a release date r_j (which may be 0), a due date (delivery date) d_j, and a positive weight w_j. The completion time of order j, denoted as C_j, is the time at which all product types for order j have been completed. If C_{ij} denotes the completion time of the items of type i for order j (on machine i), then it is clear that

$$C_j = \max(C_{1j}, \ldots, C_{mj}).$$

Several objectives are of interest, namely, the makespan $C_{\max}$, the total weighted completion time $\sum w_j C_j$ and the total weighted tardiness $\sum w_j T_j$, where the tardiness T_j is defined as $\max(C_j - d_j, 0)$. In this paper, we focus mainly on the objective of minimizing $\sum C_j$.

This model has a large number of practical applications. Most Make-To-Order environments at production facilities with a number of different resources in parallel give rise to a model of the type described above. Consider, for example, a converting operation in a process industry, such as a finishing operation that produces paper products of various different sizes (the raw material being rolls of paper). Any given type of product is produced on a specific machine. Orders arrive from clients and each order asks for certain quantities of the various different types of paper products. Each client wants to receive his entire order in one shipment (in order to minimize transportation costs and handling costs). Clearly, the model is applicable to any setting with a collection of dedicated production resources that manufacture various different products and clients who order various different products at the same time.

Yang (1998) described a different example. Consider a car repair shop. Suppose each car has several broken parts that need to be fixed. Each broken part can only be fixed by a certain mechanic in the shop. Several mechanics can work on the different parts of a car at the same time. A car leaves the shop when every broken part is fixed. This equipment maintenance and repair problem is, of course, not only applicable to cars. It is also applicable to airplane maintenance and ship repair. This equipment maintenance and repair problem is a perfect example of the model considered in this paper.

The models considered in this paper can also be regarded as a variation of the classical open shop models, in which each job consists of m operations; each one of the operations has to be done on a specific one of the m machines. The difference between the class of models considered in this paper and the classical open shop models is that in the open shop two operations that belong to the same job (i.e., order) are not allowed to be processed at the same time (even though they are processed on different machines). In the models considered in this paper two operations that belong to the same job may be processed at the same time. For these reasons, the models considered here may also be referred to as open shop models with job overlap or as concurrent open shops.

We propose the following notation for the class of scheduling problems considered in this paper. Our notation is an extension of the $\alpha \mid \beta \mid \gamma$ notation introduced by Graham, Lawler, Lenstra, and Rinnooy Kan (1979). In what follows, this parallel machine environment with fully dedicated machines is denoted by *PDm*. We assume that we have n orders and m product types (which have a one-to-one correspondence to the m machines). So, for example, $PDm \mid r_j \mid \sum T_j$ refers to the case with m machines in parallel, n different orders with order j having a release date r_j and a due date d_j. The objective is the minimization of the total tardiness. As another example, $PD6 \mid r_j, prmt \mid \sum w_j C_j$ refers to an environment with six dedicated machines in parallel. Order j has a release date r_j, and preemptions are allowed. The objective is the minimization of the total weighted completion time.

The models described above have been the focus of a fair amount of research. Wagneur and Sriskandarajah (1993) have considered the two machine open shop model with job overlap, i.e., $PD2 \parallel \sum C_j$, and presented a proof claiming that the minimization of the total completion time is strongly NP-Hard. Unfortunately, their proof is not correct (Leung et al., 2005). Independently, Sung and Yoon (1998) showed that the two machine open shop with job overlap and the total *weighted* completion time as objective is strongly NP-Hard. One of the main results in this paper is a proof that the $PDm \parallel \sum C_j$ problem is strongly NP-hard for every $m \geq 3$.

Several heuristics have been proposed in the literature for the $PDm \parallel \sum C_j$ problem. Wang and Cheng (2003) analyzed three greedy heuristics whose worst-case bounds are m. Two of these heuristics were generalizations of heuristics proposed by Sung and Yoon (1998) for two machines. In this paper, we introduce two new heuristics for $PDm \mid \sum C_j$. For one of these two heuristics, we obtain also a worst-case bound of m. Experimental results show that our heuristics consistently outperform the three heuristics that have appeared in the literature.

This paper is organized as follows. The next section presents some preliminary results for a fairly general class of objective functions that includes the total completion time as well as the total tardiness. Section 3 shows that $PDm \parallel \sum C_j$ is NP-hard in the strong sense for every fixed $m \geq 3$. Section 4 gives a description and a preliminary analysis of two new heuristics. Section 5 presents an empirical analysis of various heuristics for the total completion time objective. Finally, Section 6 discusses some extensions and presents some concluding remarks.

2. PRELIMINARY RESULTS

There are m machines and m product types; each machine can produce only one type. Since machine i is the only machine that can produce type i and type i is the only type that can be produced on machine i, the subscript i refers to a machine as well as to a product type.

Each machine is available from time 0 and produces its product as long as there is a demand. The main issue is the sequence in which the different orders are processed on the various machines.

The problem $PD1 \mid \beta \mid \gamma$ is identical to the problem $1 \mid \beta \mid \gamma$. So $PD1 \parallel \sum T_j$, $PD1 \mid r_j \mid \sum C_j$, and $PD1 \parallel \sum w_j U_j$ (where $U_j = 1$ if $C_j > d_j$ and 0 otherwise) are all NP-hard, see Du and Leung (1990) and Lenstra (1977). In this paper we therefore consider $m \geq 2$.

We start out with some general properties of optimal schedules. Assume that order j is subject to an arbitrary cost function $f_j(C_j)$, where $f_j(C_j)$ is increasing in C_j for each j. In this section we consider the problem $PD \parallel \sum f_j(C_j)$, where PD means that the number of machines m is arbitrary. This class of objective functions includes $\sum w_j C_j$, $\sum w_j T_j$, and $\sum w_j U_j$.

The following general properties can be shown fairly easily.

Lemma 2.1.

(i) *If $f_j(C_j)$ is increasing in C_j for all j, then there exists an optimal schedule for the objective function $f_{\max}$ as well as an optimal schedule for the objective function $\sum f_j(C_j)$ in which all machines process the orders in the same sequence.*

(ii) *If for some machine i there exists a machine k such that $p_{ij} \leq p_{kj}$ for $j = 1, \ldots, n$, then machine i does not play any role in determining the optimal schedule and may be ignored.*

Some remarks with regard to these properties are in order. The first property does not hold for the more general problem in which the function $f_j(C_j)$ is not monotone (e.g., problems that are subject to earliness and tardiness penalties). The second property is useful for reducing the size of the problem.

Consider the problem $PD \mid \beta \mid \sum f_j(C_j)$. Since this problem is strongly NP-hard, it is advantageous to develop dominance conditions or elimination criteria.

Lemma 2.2. If in the problem $PD1 \parallel \sum f_j(C_j)$ there are two jobs j and k such that $p_j \leq p_k$ and

$$\frac{df_j(t)}{dt} \geq \frac{df_k(t)}{dt} \quad \text{for all } t \geq 0,$$

then there exists an optimal schedule in which job j precedes job k.

Proof. The proof is by contradiction. Suppose job j is processed after job k (see Figure 1). In between jobs k and j there are a number of jobs that are being processed. Assume that in this original schedule the completion of job j is denoted by C_j and the completion of job k is denoted by C_k. Consider the following interchange: Put job j in the position of job k and vice versa. Let the completion time of job j in this new schedule be denoted by C'_j and the completion of job k by C'_k. In the new schedule, all the jobs scheduled in between the two jobs are completed earlier, since job j is shorter than job k. Since the cost functions are all monotonically increasing, the contribution of these jobs to the overall objective goes down. It is clear that $C_j = C'_k$. It is also clear that $C'_j \leq C_k$. So the move of job k increases the objective function by an amount $f_k(C'_k) - f_k(C_k)$, whereas the

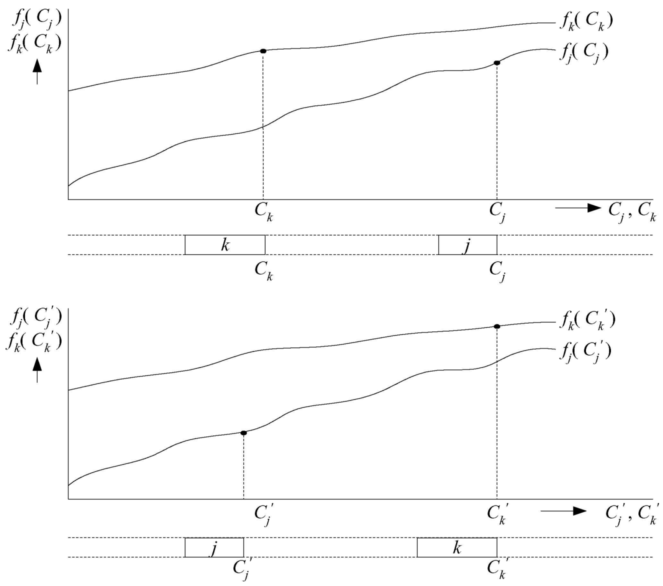

Figure 1. Illustrating the proof of Lemma 2.2.

move of job j reduces the objective function by an amount $f_j(C_j) - f_j(C_j')$. Since

$$C_j - C_j' \geq C_k' - C_k$$

and

$$\frac{df_j(t)}{dt} \geq \frac{df_k(t)}{dt}$$

the savings incurred by processing job j earlier are larger than the cost incurred by processing job k later. ■

The result in Lemma 2.2 is in a sense a generalization of the well-known dominance result of Emmons (1969). Intuitively, it means that the cost function of the shorter job j has to be "steeper" than the cost function of the longer job k; the actual levels of the costs are of no importance (see Figure 1). A dual of the result in Lemma 2.2 can be established for the single machine problem in which the jobs are subject to earliness penalties.

Lemma 2.2 can be generalized to a result for the problem $PD \parallel \sum f_j(C_j)$.

Lemma 2.3. If in the problem $PD \parallel \sum f_j(C_j)$ *there are two orders j and k such that* $p_{ij} \leq p_{ik}$ *for all i, and*

$$\frac{df_j(t)}{dt} \geq \frac{df_k(t)}{dt} \quad \text{for all } t \geq 0,$$

then there exists an optimal schedule in which order j precedes order k.

3. COMPLEXITY RESULT

Consider the special case of one of the models discussed in the previous section with the total weighted completion time objective, i.e., the problem $PD \parallel \sum w_j C_j$. Sung and Yoon (1998) have shown that when the objective is $\sum w_j C_j$, the problem becomes strongly NP-hard with two machines.

The total completion time objective $\sum C_j$ has been considered before, but has remained up to now an open problem. We present the following result.

Theorem 3.1. The problem $PD3 \parallel \sum C_j$ is strongly NP-hard.

However, the complexity of $PD2 \parallel \sum C_j$ remains an open problem. We will prove Theorem 3.1 in the remainder of this section by reducing the Numerical Matching with Target Sums (NMTS) problem to the decision version of $PD3 \parallel \sum C_j$. The Numerical Matching with Target Sums (NMTS) problem is known to be strongly NP-hard (Garey and Johnson, 1979) and can be stated as follows:

Definition 3.1 (NMTS). Let $A = \{a_1, \dots, a_n\}$, $B = \{b_1, \dots, b_n\}$, and $C = \{c_1, \dots, c_n\}$ be three sets of natural numbers. Is it possible to find a partition of $A \cup B$ into n subsets $\{a_{i(r)}, b_{j(r)}\}$ such that $c_r = a_{i(r)} + b_{j(r)}$ for $r = 1, \dots, n$?

Given any instance of NMTS, we construct an instance of the decision version of $PD3 \parallel \sum C_j$ with orders $J_a(J_b, J_c)$ of type a (respectively, b, or c) that have the following processing times on the three machines, $j = 1, 2, \dots, n$:

$$J_{a_j} : (2L + a_j, 2L + X - a_j, 0),$$

$$J_{b_j} : (2L + 2X + b_j, L + X - b_j, 0), \text{ and}$$

$$J_{c_j} : (L, 2L + 2c_j, 5L + 2X + c_j), \text{ where } \quad L > nX > 2n \sum_{j=1}^{n} c_j.$$

Since it is necessary that $\sum_{j=1}^{n} c_j = \sum_{j=1}^{n} (a_j + b_j)$ for any instance of NMTS in order to have a "Yes" answer, we assume that the given instance of NMTS satisfies this condition. Otherwise, we can map it into a "No" instance of the scheduling problem. The threshold for the decision version of $PD3 \parallel \sum C_j$ is given as:

$$D = \frac{n(15n + 7)}{2} L + (3n^2 + 2n)X + 2\sum_{j=1}^{n} c_j - \sum_{j=1}^{n} a_j + 3\sum_{j=1}^{n-1} (n - j)c_j,$$

where $c_j \le c_2 \le \cdots \le c_n$.

Lemma 3.2. If the given instance of NMTS has a solution, then there is a schedule with $\sum C_j \le D$.

Proof. Let $c_r = a_{i(r)} + b_{j(r)}$ for $1 \le r \le n$. We can construct a schedule S with the following sequence:

$$J_{a_{i(1)}}, J_{b_{j(1)}}, J_{c_1}, J_{a_{i(2)}}, J_{b_{j(2)}}, J_{c_2}, \dots, J_{a_{i(n)}}, J_{b_{j(n)}}, J_{c_n}.$$

It is easy to show that S has $\sum C_j$ exactly equal to D. ■

The following lemma is the key to the proof of Theorem 3.1.

Lemma 3.3. There is an optimal schedule in which the orders are scheduled in the order (*abc*), *repeated n times*.

Because of space constraints, we omit the proof of Lemma 3.3. It is described in Li (2005) and is also available from http://web.njit.edu/~leung/pdsumcj/proof.ps.

Lemma 3.4. There is an optimal schedule in which all orders of type a finish on machine 2, *while all orders of type b finish on machine* 1.

Proof. Let S be an optimal schedule. By Lemma 3.3, we may assume that the orders are scheduled in the order (*abc*), repeated n times. Thus, an order of type a is in position $3k+1$ and an order of type b is in position $3k+2$, $0 \le k \le n-1$. Let C_{3k+1} and C_{3k+2} be the finishing times of the orders in positions $3k+1$ and $3k+2$, respectively. By reindexing the orders if necessary, we may assume that the orders in positions $3k+1$ and $3k+2$ are $J_{a_{k+1}}$ and $J_{b_{k+1}}$, respectively, $0 \le k \le n-1$. Then, we have

$$C_{3k+1} = k(5L+2X) + 2L + \max\left\{ \begin{array}{c} \sum_{j=1}^{k}(a_j + b_j) + a_{k+1}, \\ \sum_{j=1}^{k}(2c_j - a_j - b_j) + X - a_{k+1} \end{array} \right\}$$

$$= k(5L+2X) + 2L + X + \sum_{j=1}^{k}(2c_j - a_j - b_j) - a_{k+1}. \tag{1}$$

$$C_{3k+2} = k(5L+2X) + 3L + 2X + \max\left\{ \begin{array}{c} \sum_{j=1}^{k}(a_j + b_j) + L + a_{k+1} + b_{k+1}, \\ \sum_{j=1}^{k}(2c_j - a_j - b_j) - a_{k+1} - b_{k+1} \end{array} \right\}$$

$$= k(5L+2X) + 3L + 2X + \sum_{j=1}^{k+1}(a_j + b_j) + L. \tag{2}$$

(1) and (2) follow from our assumption that $L > nX > 2n\sum_{j=1}^{n} c_j$.

Thus, the order of type a in the $(3k+1)$st position finishes on machine 2 and the order of type b in the $(3k+2)$nd position finishes on machine 1. ■

Lemma 3.5. If there is a schedule S with $\sum C_j \le D$, *then there is a solution to the given instance of NMTS*.

Proof. By the above two lemmas, the finishing times in S can be computed as follows. (For convenience, we assume that S is

$$a_1 b_1 c_1 a_2 b_2 c_2 \ldots a_n b_n c_n;$$

otherwise, we can reindex the orders.)

$$C_{3k+1} = \sum_{j=1}^{k}((2L+X-a_j)+(L+X-b_j)+(2L+2c_j))+(2L+X-a_{k+1})$$

$$= k(5L+2X)+2L+X-a_{k+1}+\sum_{j=1}^{k}(2c_j-a_j-b_j). \tag{3}$$

$$C_{3k+2} = \sum_{j=1}^{k}((2L+a_j)+(2L+2X+b_j)+L)+(2L+a_{k+1})+(2L+2X+b_{k+1})$$

$$= k(5L+2X)+4L+2X+\sum_{j=1}^{k+1}(a_j+b_j). \tag{4}$$

$$C_{3k} = \max\left\{\begin{array}{c}\sum_{j=1}^{k}((2L+a_j)+(2L+2X+b_j)+L),\\ \sum_{j=1}^{k}((2L+X-a_j)+(L+X-b_j)+(2L+2c_j))\\ \sum_{j=1}^{k}(5L+2X+c_j)\end{array}\right\}$$

$$= k(5L+2X)+\sum_{j=1}^{k}(a_j+b_j)+\max\left\{0, 2\sum_{j=1}^{k}(c_j-a_j-b_j), \sum_{j=1}^{k}(c_j-a_j-b_j)\right\}$$

$$= k(5L+2X)+\sum_{j=1}^{k}(a_j+b_j)+2\max\left\{0, \sum_{j=1}^{k}(c_j-a_j-b_j)\right\}. \tag{5}$$

To simplify notation, let

$$\delta_j = c_j-(a_j+b_j), \tag{6}$$

$$\lambda_k = \max\left\{\sum_{j=1}^{k}\delta_j, 0\right\}, \tag{7}$$

and

$$\Sigma_a = \sum_{k=0}^{n=1} C_{3k+1}, \quad \Sigma_b = \sum_{k=0}^{n-1} C_{3k+2}, \quad \Sigma_c = \sum_{k=1}^{n} C_{3k}. \tag{8}$$

We now have

$$\Sigma_a = \sum_{k=0}^{n-1} C_{3k+1} = \frac{5n^2-n}{2}L+n^2X-\sum_{j=1}^{n}a_j+\sum_{k=1}^{n-1}\sum_{j=1}^{k}(2c_j-a_j-b_j)$$

$$= \frac{5n^2-n}{2}L+n^2X-\sum_{j=1}^{n}a_j+\sum_{j=1}^{n-1}(n-j)(c_j+\delta_j). \tag{9}$$

$$\Sigma_b = \sum_{k=0}^{n-1} C_{3k+2} = \frac{5n^2+3n}{2} L + n(n+1)X + \sum_{j=1}^{n} (n+1-j)(c_j - \delta_j). \tag{10}$$

$$\Sigma_c = \sum_{k=1}^{n} C_{3k} = \frac{5n(n+1)}{2} L + n(n+1)X + \sum_{j=1}^{n} (n+1-j)(c_j - \delta_j) + 2\sum_{k=1}^{n} \lambda_k. \tag{11}$$

Therefore, the total completion time in S is

$$\begin{aligned}\sum_{j=1}^{3n} C_j &= \Sigma_a + \Sigma_b + \Sigma_c \\ &= \frac{n(15n+7)}{2} L + (3n^2+2n)X + 2\sum_{j=1}^{n} c_j - \sum_{j=1}^{n} a_j \\ &\quad + 3\sum_{j=1}^{n-1} (n-j)c_j + 2\sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{n} (n+2-j)\delta_j \\ &= D + 2\sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{n} (n+2-j)\delta_j. \end{aligned} \tag{12}$$

Since the total completion time in S is no larger than D, we must have

$$2\sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{n} (n+2-j)\delta_j \le 0.$$

We now show that this implies that $\delta_j = 0$ for all $1 \le j \le n$, which means that the instance of NMTS has a solution. We first observe that

$$\sum_{j=1}^{n} c_j = \sum_{j=1}^{n} (a_j + b_j) \Rightarrow \sum_{j=1}^{n} \delta_j = \sum_{j=1}^{n} ((c_j - (a_j + b_j)) = 0. \tag{13}$$

Now,

$$\begin{aligned} 2\sum_{j=1}^{n} \lambda_j - \sum_{j=1}^{n} (n+2-j)\delta_j &= 2\sum_{j=1}^{n} \max\left\{0, \sum_{i=1}^{j} \delta_i\right\} - \sum_{j=1}^{n} (n+2-j)\delta_j \\ &= 2\max\left\{0, \delta_1, \delta_1+\delta_2, \ldots, \sum_{j=1}^{n} (n+1-j)\delta_j\right\} - \sum_{j=1}^{n} (n+2-j)\delta_j. \\ &= 2\max\left\{0, \delta_1, \delta_1+\delta_2, \ldots, \sum_{j=1}^{n} (n+1-j)\delta_j\right\} - \sum_{j=1}^{n} (n+1-j)\delta_j \end{aligned} \tag{14}$$

(recall that $\sum_{j=1}^{n} \delta_j = 0$).

We now have to show that

$$2\max\left\{0, \delta_1, \delta_1+\delta_2, \ldots, \sum_{j=1}^{n}(n+1-j)\delta_j\right\} \le \sum_{j=1}^{n}(n+1-j)\delta_j$$

implies that $\delta_j = 0$ for each $1 \le j \le n$.

We will show this by contradiction. Suppose it is not the case; that is, one or more δ_j are strictly positive and one or more are strictly negative.

We first show that it is impossible for a partial sum $\sum_{j=1}^{k}\delta_j$ to be greater than zero. If such a partial sum is positive, then the left hand side (LHS) must be greater than 0. If the right hand side (RHS) is negative, the contradiction is immediately established; if the RHS is positive, then the LHS is at least twice as large because of the last term on the LHS and the contradiction is also established. So a partial sum can never be strictly positive.

Consider the case where $\sum_{j=1}^{k}\delta_j$ are always either negative or zero. If all the partial sums are negative or zero, then $\sum_{j=1}^{n}(n+1-j)\delta_j$ must be negative. This follows from the fact that for every δ_l that is positive by a certain amount, there is(are) one or more δ_j that are negative by that amount with $j < l$. From the fact that the multiplier in the sum on the RHS is larger for $j < l$, it follows that the sum on the RHS is negative. With the LHS of the inequality being 0 and the RHS negative, a contradiction is again established. ■

Because of the complexity of the problem with the total completion time objective, it appears to be of interest to study heuristics which are based on simple rules.

4. HEURISTICS

Various heuristics that appear attractive in dealing with the total completion time objective have already been proposed in the literature. Most of these heuristics are greedy heuristics and generate a sequence of orders progressively one at a time. The first three heuristics mentioned below have been studied in the literature; the last two are new.

Definition 4.1. The Shortest Total Processing Time first (STPT) heuristic generates a sequence of orders one at a time, each time selecting as the next order the one with the smallest total amount of processing over all m machines.

Definition 4.2. The Shortest Maximum Processing Time first (SMPT) heuristic generates a sequence of orders one at a time, each time selecting as the next order the one with the smallest maximum amount of processing on any one of the m machines.

Definition 4.3. The Smallest Maximum Completion Time first (SMCT) heuristic first sequences the orders in nondecreasing order of p_{ij} on each machine $i = 1, 2, \ldots, m$ then computes the completion time for order j as $C'_j = \max_{i=1}^{m}\{C_{ij}\}$, and finally schedules the orders in nondecreasing order of C'_j.

Definition 4.4. The Shortest Processing Time first applied to the machine with the largest load (SPTL) is a heuristic that generates a sequence of orders one at a time, each time selecting as the next order the one with the smallest processing time on the machine that currently has the largest load.

Definition 4.5. The Earliest Completion Time first (ECT) heuristic generates a sequence of orders one at a time; each time it selects as the next order the one that would be completed the earliest.

The SMPT and the STPT heuristics have been analyzed by Sung and Yoon (1998). Besides the SMPT and the STPT heuristics, Wang and Cheng (2003) also studied the SMCT heuristic, which is somewhat similar but not the same as the ECT heuristic we propose above. It seems that the SPTL heuristic is new.

That such greedy heuristics or priority rules may perform poorly is well-known. The following examples illustrate instances in which three of the heuristics described above do not perform well.

Example 4.1. Consider two machines and n orders that require 1 time unit on machine 1 and 0 on machine 2, n orders that require 1 time unit on machine 2 and 0 on machine 1, and $a \times n$ jobs that require $1 - \epsilon$ on machine 1 and $1 - \epsilon$ on machine 2. The ratio

$$\frac{\sum C_j(\text{ECT})}{\sum C_j(\text{OPT})}$$

is increasing in n and when $n \to \infty$ it becomes

$$\frac{a^2 + 4a + 2}{a^2 + 2a + 2}$$

This ratio reaches its maximum of $\sqrt{2}$ when $a = \sqrt{2}$. The optimal schedule in this case is to schedule the orders with the smallest total processing time first, i.e., the schedule is generated according to the STPT rule.

It is clear how to generalize this counterexample to an arbitrary number of machines m. The ECT rule will then perform even worse in comparison with the optimal rule. A counterexample to the ECT rule can also be constructed when all orders require the same total amount of processing. It should be noted that Example 4.1 also applies to the SMCT heuristic.

A counterexample to the STPT rule can be constructed as follows.

Example 4.2. Consider 2 machines and n orders that require 1 unit on machine 2 and ϵ on machine 1, and $a \times n$ orders that require 1 unit on machine 1 only. By taking $a = (1 + \sqrt{5})/2$, we obtain a value of the ratio of 1.618.

This counterexample can be generalized to an arbitrary number of machines m. The STPT rule will then perform even worse in comparison with the optimal rule. Sung and Yoon (1998) proved that for two machines

$$\frac{\sum C_j(\text{STPT})}{\sum C_j(\text{OPT})} \leq 2.$$

Wang and Cheng (2003) generalized the above result to m machines and the bound is m. So, the worst-case ratio of the STPT rule for two machines is bounded above by 2 and Example 4.2 shows that the ratio can be as high as 1.618.

The performance of SPTL can be very bad. Consider the following example.

Example 4.3. Consider two machines and one order that requires 0 time on machine 1 and A units of time on machine 2; in addition, there are B orders that require 1 time unit on both machines. If we consider at $t = 0$ machine 1, and let $A \gg B$, then the ratio becomes $(B+1)$, which can be arbitrarily large. However, if at $t = 0$ we consider machine 2, then the result is optimal.

Theorem 4.1. For $PDm \parallel \sum C_j$,

$$\frac{\sum C_j(\text{ECT})}{\sum C_j(\text{OPT})} \leq m.$$

Proof. Let $p_j = \max_{1 \leq i \leq m}\{p_{ij}\}$; let S_{ECT} denote the schedule generated by ECT and let S_{OPT} denote an optimal schedule. Furthermore, without loss of generality, we may assume that the orders are labelled in such a way that

$$p_1 \leq p_2 \leq \cdots \leq p_n.$$

First, we want to show that

$$C_j(S_{\text{ECT}}) \leq \sum_{k=1}^{j} p_k, \quad j = 1, 2, \ldots, n. \tag{15}$$

Clearly, the first order scheduled in S_{ECT} must be the one with the shortest maximum processing time. Thus,

$$C_1(S_{\text{ECT}}) = p_1.$$

Suppose that there exists a smallest position $j^*(1 < j^* < n)$ of the orders scheduled in S_{ECT}, such that

$$C_{j^*-1}(S_{\text{ECT}}) \leq \sum_{k=1}^{j^*-1} p_k,$$

but

$$C_{j^*}(S_{\text{ECT}}) > \sum_{k=1}^{j^*} p_k.$$

Then,

$$C_{j^*}(S_{\text{ECT}}) - C_{j^*-1}(S_{\text{ECT}}) > p_{j^*}.$$

So the maximum processing time of the (j^*)th order in S_{ECT} is larger than p_{j^*}. It follows that there exists at least one order with a subscript $l(2 \leq l \leq j^*$ and $p_l \leq p_{j^*})$ that was scheduled after position j^* in S_{ECT}. However, if we schedule order l in position j^* in S_{ECT}, then $C_{j^*}(S_{\text{ECT}})$ will be smaller. This leads to a contradiction, since the ECT heuristic always chooses as the next order the one that would be completed the earliest.

From (15), we have

$$\sum_{j=1}^{n} C_j(S_{\text{ECT}}) \leq \sum_{j=1}^{n}\sum_{k=1}^{j} p_k. \tag{16}$$

Now, let $[j]$ be the order position in S_{OPT}. Then,

$$\begin{aligned} C_{[j]}(S_{\text{OPT}}) &= \max_{l\leq i\leq m}\left\{\sum_{k=1}^{j} p_{i[k]}\right\} \geq \sum_{k=1}^{j}\left(\sum_{i=1}^{m} p_{i[k]}\right)\Big/ m \\ &\geq \sum_{k=1}^{j}\max_{1\leq i\leq m}\left\{p_{i[k]}\right\}\big/ m. \end{aligned} \tag{17}$$

It follows that

$$\sum_{j=1}^{n} C_j(S_{\text{OPT}}) \geq \sum_{j=1}^{n}\sum_{k=1}^{j}\max_{1\leq i\leq m}\{p_{i[k]}\}/m \geq \sum_{j=1}^{n}\sum_{k=1}^{j} p_k/m. \tag{18}$$

The last inequality is due to Smith's WSPT rule (Smith, 1956).

From (16) and (18), the result follows. ■

5. EMPIRICAL ANALYSIS OF THE HEURISTICS

In this section we describe an empirical analysis of the heuristics for the $PDm \parallel \sum C_j$ problem. Due to the NP-hardness of the problem for $m \geq 3$ ($m = 2$ remains open), they are not likely to produce optimal solutions for the problem within limited running time. Thus, to evaluate the performance of these heuristics, we apply a Tabu Search routine (Glover, 1997) in an attempt to improve on the results obtained by these heuristics; the Tabu Search routine basically serves as a tool for measuring the effectiveness of the five heuristics that are being analyzed.

5.1. Schedule generation by heuristics

Each one of the five heuristics is applied to every instance being considered and the five schedules generated are compared with one another. Implementing the five heuristics is relatively easy. Through a sorting algorithm, both STPT and SMPT can be implemented to run in $O(mn + n\log n)$ time, and SMCT can be implemented to run in $O(mn\log n)$ time. Both SPTL and ECT can be implemented in a natural way to run in $O(mn^2)$ time. Each heuristic produces a schedule and we feed the best one of the five schedules generated into the Tabu Search for postprocessing.

5.2. Postprocessing with Tabu Search

5.2.1. Tabu structure

In many applications of Tabu Search, reverse moves resulting in recently visited solutions are prohibited. However, if we use a short-term tabu list, it may be hard to avoid cycling. Surveys on the application of Tabu Search in intelligent scheduling systems can be found in Zweben and Fox (1998) and Barnes, Laguna and Glover (1995).

In this section, we propose a tabu structure using long-term memory; see Glover and Laguna (1997). The idea is to encode the solutions in an informative but concise way so that the representations describe the solution accurately and can be stored in a tabu list without taking too much memory. In addition, the representations are kept throughout until the Tabu Search procedure stops.

To represent a solution in a concise way, we encode a schedule using a data structure that consists of four data fields: (1) Cost of the schedule; (2) index of the first order in the sequence; (3) index of the middle order in the sequence; (4) index of the last order in the sequence.

It is possible that two different schedules have the same four data entries, but the probability of this occurring is small. If the four data entries of two different schedules are the same, then the two schedules are considered the same. Using such a representation, cycling definitely will not occur, but there is a small probability that some solutions will not be visited at all.

It should be noted that our settings are somewhat similar to the tabu cycle method (Glover and Laguna, 1997), which is based on short-term memory. For an implementation of the tabu cycle method, the reader is referred to Laguna (2004).

5.2.2. Neighborhood generating mechanism

A neighborhood generating mechanism defines a set of solutions that are to explored by a local search procedure imbedded in Tabu Search.

Given a schedule S as shown in Figure 2(a), a so-called *Adjacent Subsequence Interchange* (ASI) as described in Figure 2(b) generates a move from S to a neighboring solution of S. A so-called *Subsequence Reversal and Interchange* (SRI) also generates a move to a neighboring solution, see Figure 2(c).

Each move is specified by three parameters: k_1, k_2, and k_3, where the positions k_1 and k_2 are the start and end positions of the segment that has to be transposed, while position k_3, which lies outside the segment $[k_1, k_2]$, is the new position in front of which the segment has to be moved to.

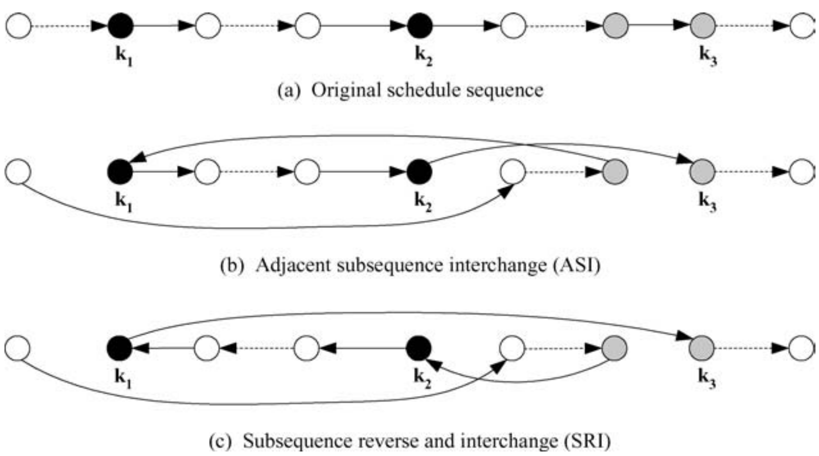

Figure 2. Neighborhood generating operators.

Thus, these three position parameters define a neighborhood with about $O(n^3)$ distinct solutions for both ASI and SRI. To make the neighborhood size smaller so that the tabu search will run faster, we may impose $k_2 - k_1 \leq 6$. Thus, the neighborhood size becomes $O(n^2)$. For any schedule S, we denote $N_{\text{ASI}}(S)$ and $N_{\text{SRI}}(S)$ as the neighborhoods generated by ASI and SRI, respectively.

The dominance result described in Lemma 2.3 is incorporated in the generation of $N_{\text{ASI}}(S)$ and $N_{\text{SRI}}(S)$ in order to prune the moves that result in solutions that we know can never be optimal. Thus, the number of solutions in $N_{\text{ASI}}(S)$ and $N_{\text{SRI}}(S)$ to be considered can be reduced.

5.2.3. Tabu Search procedure

The Tabu Search operates as follows.

Step 1 {*Initialization*}.
Select the best schedule S generated by STPT, SMPT, SMCT, SPTL, and ECT.
Let $S_b \leftarrow S$.
Set tabu list Γ to be empty.
Step 2 {*Tabu Search*}.
Explore all the solutions in $N_{\text{ASI}}(S)$ and $N_{\text{SRI}}(S)$, and choose the best solution S_b' that is not in tabu list Γ.
If S_b' is better than S_b, then let $S_b \leftarrow S_b'$.
Add S into tabu list Γ.
Let $S \leftarrow S_b'$.
Step 3 {*Output*}.
If stop criterion is not met yet, goto Step 2. Otherwise, output S_b.

In our experiment, the stopping criterion is 1000 nonimproving iterations in Step 2. We did not incorporate the aspiration criterion and intensification/diversification mechanisms in the tabu search, since we just need to evaluate the quality of the results produced by the heuristics. Using these more intelligent mechanisms, the tabu search would have performed better, but this is not the focus of this paper. Readers who are interested in these techniques are referred to Glover and Laguna (1997).

The running time of the Tabu Search algorithm is dominated by Step 2, each iteration of which examines $O(n^2)$ solutions in the neighborhoods $N_{\text{ASI}}(S)$ and $N_{\text{SRI}}(S)$. Since each exploration takes $O(mn)$ time to compute the objective cost of a solution and $O(1)$ time to check if a solution is in the tabu list Γ, each iteration of Step 2 takes $O(mn^3)$ time.

5.3. Generation of problem instances

For each problem size with $n = 20, 50, 100, 200$ orders and $m = 2, 5, 10, 20$ machines, 30 instances are randomly generated using a factor called *order diversity*. The order diversity k is used to characterize the number of product types each order requires. The following three cases of order diversity are considered:

$k = 2$: In problem instances 1–6 each order requests two different product types.
$k = m$: In problem instances 7–12 each order requests the maximum number of different product types, namely m; m is the number of machines.
$k = r$: In problem instances 13–30 each order requests a random number (r) of different product types; r is randomly generated from the uniform distribution $[1, m]$.

Table 1. The percentage that each heuristic performs the best

Heuristic	Percentage
STPT	6.0
SMPT	0
SMCT	0
SPTL	6.0
ECT	88.0

When the number of product types, l, for each order j is determined, l machines are chosen randomly. For each machine i that is selected, an integer processing time p_{ij} is generated from the uniform distribution [1, 100]. In total, $4 \times 4 \times 30 = 480$ instances are generated.

5.4. *Experimental results and analysis*

The algorithms are implemented in C++. The running environment is based on the RedHat Linux 7.0 operating system; the PC used was a Pentium II 400 Mhz with 128 MB RAM.

The detailed results are shown in Tables A1–A8 which appear in Appendix A. In each one of the tables, the results concerning the five heuristics appear in columns 3–7. The remaining columns have:

m: The number of machines.
$H_{\min}$: The name of the heuristic that yields the best result.
TS: The result obtained with Tabu Search.
Imp: The improvement obtained with Tabu Search (as a percentage).
T_{h}(sec): The total running time (in seconds) of the five heuristics.
T_{ts}(sec): The running time (in seconds) of Tabu Search.

Table 1 aggregates the results from Tables A1–A8 in the Appendix. It shows that ECT is the best heuristic. Both STPT and SPTL produce the best schedule for some instances. However, neither SMPT nor SMCT produces the best schedule in any instance.

From Tables A1 to A8, we can see that all instances for which STPT performs the best occur when $m = 2$, while the instances for which SPTL performs the best are distributed over all values of m. When we have more than two machines, ECT tends to be the best, and SPTL tends to be the second best. The differences between ECT and the other heuristics, especially the first three, can be quite substantial (often more than 5%; especially when the number of machines is large).

The average costs in Tables 2–4, respectively, show that for $k = 2$, the objective function decreases when m increases; for $k = m$, the objective function increases when m increases; for $k = r$, the change in the objective costs is not so clear.

With regard to the average performance, Tables 2–4 also show that ECT performs better than all other heuristics. The improvement obtained by the Tabu Search is not that much. The tables also reveal that SMPT is the worst heuristic, while SMCT is slightly better than SMPT. STPT and SPTL are better than SMPT and SMCT. Therefore, the average performance is also consistent with Table 1.

Table 2. Comparison of average costs when $k = 2$

n	m	STPT	SMPT	SMCT	SPTL	ECT	TS	Imp
20	5	5140	5252	4795	4754	4539	4389	3.35
	10	3904	4082	3408	3622	3161	3118	1.38
	20	3246	3156	2543	2809	2411	2401	0.43
50	5	26962	28081	25002	24315	23193	22831	1.58
	10	16684	17079	14291	15396	12913	12808	0.83
	20	11455	11621	9356	10953	8447	8431	0.20
100	5	97028	99304	92602	89609	85675	84288	1.63
	10	60583	62130	50860	54137	46273	46209	0.13
	20	38073	38177	30852	35315	27197	27190	0.02
200	5	384440	39887	364984	350516	341139	334658	1.93
	10	212608	215531	187304	190922	170692	170547	0.08
	20	127751	127392	107108	119232	93620	93607	0.02

Table 3. Comparison of average costs when $k = m$

n	m	STPT	SMPT	SMCT	SPTL	ECT	TS	Imp
20	2	9719	9814	9662	9525	9283	9193	0.97
	5	12006	12057	11909	11662	11338	11166	1.52
	10	12934	13114	12515	12418	12031	11808	1.88
	20	13142	13410	13187	12758	12314	12031	2.37
50	2	50672	51973	51382	51386	49689	48817	1.55
	5	63865	66194	64880	61930	60249	59114	1.92
	10	68298	72787	70292	67787	65094	63865	1.93
	20	71325	73682	72242	70094	67394	66227	1.78
100	2	209003	217536	214953	209384	206885	202199	1.83
	5	245043	253867	250694	241537	232949	228461	1.95
	10	263542	274975	269076	258865	248448	245098	1.38
	20	272267	280485	275442	265886	256651	253466	1.27
200	2	827466	848915	839999	829966	820013	799041	2.22
	5	961302	1001502	987641	951628	917933	900094	1.97
	10	1004000	1062621	1043997	998574	965789	952002	1.45
	20	1038045	1092310	1071347	1026776	993572	985501	0.82

Tables 2–4 also show that, when $k = 2$, the percentage improvement obtained through Tabu Search over the best result from the five heuristics decreases if m increases; when $k = m$ or $k = r$, this percentage tends to increase when m increases (even though for large n and m Tabu Search fails to provide much of an improvement because of the large solution spaces). This is consistent with the fact that the performance ratios of the heuristics become worse when m increases.

Table 4. Comparison of average costs when $k = r$

n	m	STPT	SMPT	SMCT	SPTL	ECT	TS	Imp
20	2	6341	6500	6203	6262	5996	5887	1.58
	5	6904	6822	6416	6295	6019	5917	1.79
	10	6661	6914	6522	6064	5811	5684	2.12
	20	7456	8218	7562	6979	6704	6544	2.43
50	2	35224	36975	35722	35022	34217	33206	2.68
	5	36075	38168	36148	33181	32493	31433	3.38
	10	34765	38639	36155	32066	30864	30056	2.65
	20	35194	39876	37354	33121	31951	30910	3.29
100	2	150013	155842	151452	148438	145675	141165	3.02
	5	128211	138648	132673	120410	118702	114338	3.71
	10	124160	136125	130186	114024	111213	107923	3.03
	20	124063	141346	134570	115521	110666	108286	2.17
200	2	558391	583815	569089	561344	551590	531680	3.36
	5	469938	520905	498128	443399	445451	425430	3.99
	10	451059	511691	490323	422835	416597	405359	2.75
	20	453159	520763	497481	425083	411549	404229	1.81

6. CONCLUSIONS AND EXTENSIONS

This paper has focused on the class of objectives $\sum f_j(C_j)$, which includes various objectives such as the total weighted completion time $\sum w_j C_j$ and the total weighted tardiness $\sum w_j T_j$. The heuristics introduced are applicable to the $\sum C_j$ objective. They appear to be very effective and do give some insights that are useful for the development of heuristics for more general objective functions. For example, for the $\sum w_j C_j$ objective the following heuristic can be considered, which may be regarded as a generalization of either the ECT heuristic or the SPTL heuristic. Assume that the last order in the partial schedule that already has been fixed is completed at time t. Compute for each one of the remaining orders its completion time assuming it is the one selected to go next. If order k goes next its completion time is C_k. The order with the highest ratio of $w_k/(C_k - t)$ is selected then as the order to go next. This heuristic could be referred to as the *Weighted Earliest Completion Time first* heuristic. One direction for future research would involve a performance analysis of such a heuristic.

This paper focuses mainly on nonpreemptive scheduling problems. Of course, each nonpreemptive problem has a preemptive counterpart. Often when preemptions are allowed, the optimal schedule is still nonpreemptive. It is of interest to determine the conditions under which a nonpreemptive schedule is optimal in the class of preemptive schedules. For example, if all $r_j = 0$, and if $f_j(C_j)$ is increasing in C_j, the optimal preemptive schedule is nonpreemptive.

However, if the release dates of all the orders are different, then preemptions start playing an important role. In contrast to the nonpreemptive problems, the problems that do allow preemptions may still be easy when the orders have different release dates. If preemptions are allowed, then it can be shown that the preemptive SPT rule is optimal for $PD|r_j, prmt|\sum C_j$ under certain agreeability conditions of the processing times. The nonpreemptive heuristics described in the previous section for nonpreemptive problems with all the orders available at time 0, can be modified fairly easily to be applicable in a preemptive environment in which the orders have different release dates.

A number of complexity issues still remain open. For example, with regard to the total completion time, the complexity status of $PDm\|\sum C_j$ has not been established for $m = 2$.

It is clear that due date related objectives, e.g., $\sum w_j T_j$ and $\sum w_j U_j$, deserve additional attention. A separate paper focuses on complexity issues as well as heuristics that can be applied to problems $PDm|\beta|\sum w_j T_j$ and $PDm|\beta|\sum w_j U_j$.

The model considered in this paper is a special case of a more general class of models in which there are k different product types and product type i can be produced on any machine that belongs to a specific subset of the machines, namely subset M_i. So, a machine may be capable of producing various different product types. If a machine switches over from one product type to another product type, it may require a sequence dependent setup time. This more general model has been studied by Julien and Magazine (1990) and Yang (1998). Julien and Magazine (1990) described some applications of this more general class of models.

7. APPENDIX A: DETAILED EXPERIMENTAL RESULTS

Table A1. Comparative overview of performance of heuristics for $\sum C_j : n = 20; m = 2, 5$

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_h(sec)	T_{ts}(sec)
2	1	9394	9659	9377	9154	8925	ECT	8829	1.1	0.00	21
	2	9912	10474	10268	9698	9773	SPTL	9631	0.7	0.01	23
	3	10356	10811	10112	10113	9591	ECT	9400	2.0	0.00	21
	4	8405	8278	8257	8160	7968	ECT	7922	0.6	0.00	21
	5	8345	8311	8053	8149	7864	ECT	7860	0.1	0.00	21
	6	9172	9650	9750	9159	9044	ECT	8997	0.5	0.00	21
	7	8385	8067	8131	8261	8011	ECT	7901	1.4	0.01	21
	8	10551	10363	10569	10431	10171	ECT	10134	0.4	0.01	21
	9	7731	7805	7718	7672	7605	ECT	7564	0.5	0.01	21
	10	11215	11532	10945	10437	10356	ECT	10140	2.1	0.00	21
	11	10983	11380	10929	10830	10183	ECT	10093	0.9	0.00	21
	12	9449	9735	9682	9516	9371	ECT	9324	0.5	0.01	21
	13	5938	5975	5541	5858	5413	ECT	5400	0.2	0.00	21
	14	5017	5225	4784	4823	4617	ECT	4499	2.6	0.00	21
	15	5503	5931	5688	5695	5381	ECT	5306	1.4	0.00	21
	16	5189	5218	5141	5594	4907	ECT	4811	2.0	0.00	21
	17	10100	9923	9150	9045	8990	ECT	8964	0.3	0.00	21
	18	7025	7200	7028	7009	6741	ECT	6556	2.8	0.01	22
	19	6490	6774	6799	6531	6429	ECT	6277	2.4	0.01	21
	20	7440	7363	6907	6857	6643	ECT	6620	0.3	0.00	21
	21	7053	7319	7160	6720	6918	SPTL	6682	0.6	0.00	23

(*Continued on next page.*)

Table A1. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
	22	6579	6830	6367	6589	6088	ECT	5956	2.2	0.00	22
	23	7496	8221	7547	7583	7265	ECT	7111	2.2	0.00	21
	24	6881	6813	6596	6766	6447	ECT	6439	0.1	0.01	21
	25	5443	5380	5125	5368	5045	ECT	5031	0.3	0.00	21
	26	4526	4710	4500	4371	4191	ECT	4153	0.9	0.00	21
	27	6090	6215	5919	6050	5790	ECT	5581	3.7	0.01	21
	28	5124	5344	5201	5249	5070	ECT	4925	2.9	0.00	21
	29	6195	6146	5852	6362	5824	ECT	5753	1.2	0.01	21
	30	6047	6408	6350	6237	6175	STPT	5903	2.4	0.01	21
5	1	6359	5685	5243	5230	5201	ECT	4879	6.6	0.00	31
	2	4303	4501	4254	4185	4067	ECT	3978	2.2	0.00	24
	3	5502	6324	5129	4812	4672	ECT	4564	2.4	0.00	48
	4	4573	4685	4447	4482	4264	ECT	4076	4.6	0.00	23
	5	4447	4633	4380	4357	4070	ECT	4000	1.8	0.01	24
	6	5658	5682	5315	5460	4960	ECT	4838	2.5	0.00	27
	7	10981	11405	11029	10981	10438	ECT	10393	0.4	0.01	26
	8	12034	12059	11799	11073	11105	SPTL	10856	2.0	0.01	33
	9	12646	13079	12942	12168	12040	ECT	11796	2.1	0.00	27
	10	12743	12564	12399	12661	12036	ECT	11951	0.7	0.00	27
	11	12900	12728	12603	12291	12182	ECT	12104	0.6	0.00	29
	12	10730	10504	10682	10799	10225	ECT	9894	3.3	0.00	36
	13	9259	8106	7999	8036	7633	ECT	7477	2.1	0.01	29
	14	6023	6811	6151	5810	5608	ECT	5390	4.0	0.00	45
	15	5677	5122	5252	5201	4905	ECT	4742	3.4	0.00	31
	16	5742	5901	5814	5618	5461	ECT	5413	0.9	0.00	29
	17	6775	7259	6399	5770	5742	ECT	5599	2.6	0.01	28
	18	6491	6947	6619	6488	6348	ECT	6237	1.8	0.01	28
	19	5615	5720	5352	5245	5010	ECT	4844	3.4	0.00	27
	20	6217	5849	6058	5835	5647	ECT	5543	1.9	0.01	30
	21	10803	9924	9060	9294	8772	ECT	8697	0.9	0.01	33
	22	5613	5756	5486	5413	5167	ECT	5131	0.7	0.01	27
	23	8189	7220	7061	7022	6831	ECT	6746	1.3	0.00	42
	24	9983	10089	9085	8501	8188	ECT	8147	0.5	0.01	31
	25	5797	5509	5421	5399	5146	ECT	5047	2.0	0.01	33
	26	7886	7711	7240	7129	6790	ECT	6704	1.3	0.00	33
	27	6763	7087	6332	6199	6004	ECT	5884	2.0	0.00	29
	28	6031	6047	5328	5523	5141	ECT	5134	0.1	0.00	31
	29	5622	6077	5363	5301	4905	ECT	4836	1.4	0.01	30
	30	5784	5656	5463	5524	5037	ECT	4938	2.0	0.00	34

Table A2. Comparative overview of performance of heuristics for $\sum C_j$: $n = 20; m = 10, 20$

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_h(sec)	T_{ts}(sec)
10	1	4664	5227	3743	3952	3523	ECT	3523	0.0	0.00	31
	2	3288	3516	3092	3232	2790	ECT	2763	1.0	0.00	32
	3	4841	4199	3738	4259	3451	ECT	3342	3.3	0.00	30
	4	3250	3254	2694	3056	2616	ECT	2598	0.7	0.01	56
	5	3792	4386	3729	3854	3484	ECT	3447	1.1	0.01	33
	6	3587	3909	3452	3381	3102	ECT	3035	2.2	0.00	32
	7	13359	13735	12662	12783	12489	ECT	12144	2.8	0.01	64
	8	13028	13652	12447	12549	12213	ECT	11819	3.3	0.00	47
	9	12365	12424	12238	11911	11790	ECT	11445	3.0	0.00	48
	10	12542	13106	12250	12068	11674	ECT	11585	0.8	0.00	43
	11	12106	12115	12273	11961	11272	ECT	11159	1.0	0.01	44
	12	14206	13649	13217	13234	12749	ECT	12698	0.4	0.00	39
	13	5857	6536	6323	5915	5650	ECT	5593	1.0	0.00	51
	14	5818	6375	5830	5543	5661	SPTL	5263	5.3	0.00	34
	15	5823	5840	5417	5241	4856	ECT	4771	1.8	0.01	33
	16	7550	7631	7635	7194	6828	ECT	6688	2.1	0.01	37
	17	6232	5921	5949	5457	5160	ECT	5070	1.8	0.01	39
	18	6340	7395	7067	5844	5715	ECT	5668	0.8	0.01	48
	19	6614	7445	5727	5586	5428	ECT	5358	1.3	0.01	37
	20	6852	7145	6400	5924	5661	ECT	5574	1.6	0.01	75
	21	6441	7070	6541	6253	6027	ECT	5911	2.0	0.01	37
	22	8361	8058	8175	7261	7253	ECT	6994	3.7	0.00	42
	23	6994	7851	6811	6761	6252	ECT	6252	0.0	0.01	34
	24	9392	8367	8144	7773	7492	ECT	7279	2.9	0.01	50
	25	4196	4398	4295	4291	4044	ECT	3885	4.1	0.01	32
	26	6079	6266	6243	5709	5562	ECT	5437	2.3	0.00	33
	27	6588	7852	7234	6093	5807	ECT	5621	3.3	0.01	33
	28	5435	5943	5435	5239	4807	ECT	4802	0.1	0.00	32
	29	8270	7774	7218	6599	6319	ECT	6253	1.1	0.01	33
	30	7055	6578	6951	6474	6076	ECT	5897	3.0	0.00	35
20	1	3190	3114	2532	2846	2414	ECT	2393	0.9	0.00	65
	2	4729	4092	2858	3174	2734	ECT	2721	0.5	0.01	57
	3	2589	2648	2209	2314	2066	ECT	2041	1.2	0.01	44
	4	2509	2886	2277	2525	2177	ECT	2177	0.0	0.01	33
	5	3760	3401	2993	3474	2843	ECT	2843	0.0	0.00	37
	6	2700	2795	2389	2523	2233	ECT	2233	0.0	0.00	32
	7	13157	12390	12713	12352	11717	ECT	11578	1.2	0.00	91
	8	14013	13750	13355	13320	12881	ECT	12584	2.4	0.01	64

(*Continued on next page.*)

Table A2. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
	9	13068	13500	13078	12470	12180	ECT	11979	1.7	0.01	115
	10	13531	13888	13780	12989	12654	ECT	12448	1.7	0.01	59
	11	12362	12761	12507	12394	11765	ECT	11354	3.6	0.00	76
	12	12722	14171	13690	13025	12689	ECT	12244	3.6	0.00	84
	13	8078	8825	8226	7705	7470	ECT	7356	1.5	0.00	45
	14	9154	9385	9128	8698	8515	ECT	7984	6.7	0.01	48
	15	7907	8997	7885	7226	7188	ECT	7047	2.0	0.01	41
	16	7251	7604	7644	6773	6535	ECT	6459	1.2	0.00	42
	17	7312	9301	7936	6992	6873	ECT	6820	0.8	0.00	43
	18	5452	6154	5451	5029	4616	ECT	4532	1.9	0.01	46
	19	6614	7011	6888	6315	6226	ECT	6001	3.7	0.01	42
	20	8366	9251	8773	7974	7708	ECT	7588	1.6	0.01	43
	21	6025	6997	6122	5759	5194	ECT	5003	3.8	0.01	41
	22	5691	6213	6007	5605	5338	ECT	5165	3.3	0.00	43
	23	7445	8492	7622	6560	6351	ECT	6209	2.3	0.00	44
	24	7230	7903	7126	6856	6448	ECT	6432	0.2	0.01	42
	25	7445	8251	7580	7108	6658	ECT	6631	0.4	0.00	45
	26	8532	7849	7521	7282	6846	ECT	6767	1.2	0.00	45
	27	8370	9872	8748	8150	7733	ECT	7537	2.6	0.01	72
	28	7290	7617	6922	6391	6159	ECT	6008	2.5	0.01	41
	29	7413	8524	7907	6927	6874	ECT	6616	3.9	0.00	46
	30	8629	9685	8628	8270	7947	ECT	7632	4.1	0.01	46

Table A3. Comparative overview of performance of heuristics for $\sum C_j : n = 50; m = 2, 5$

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
2	1	55059	57124	56877	55225	55233	STPT	53943	2.1	0.00	136
	2	64070	62374	61649	61300	60232	ECT	59333	1.5	0.00	139
	3	56442	59004	58492	57849	57039	STPT	55667	1.4	0.01	149
	4	51924	52572	52163	51486	50996	ECT	49920	2.2	0.00	177
	5	50350	51255	50159	50237	48119	ECT	47586	1.1	0.00	290
	6	54309	56097	56501	54753	53825	ECT	52978	1.6	0.01	141
	7	51163	52608	51649	51241	50216	ECT	49300	1.9	0.00	150
	8	54104	56537	55719	55324	54258	STPT	52839	2.4	0.01	147
	9	49520	51916	51335	50329	48814	ECT	48161	1.4	0.01	143
	10	51727	51349	50765	51864	48824	ECT	48120	1.5	0.01	204
	11	48595	50335	50100	50192	49149	STPT	48108	1.0	0.00	175
	12	48925	49092	48726	49365	46875	ECT	46351	1.1	0.01	221

(*Continued on next page.*)

Table A3. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_h(sec)	T_{ts}(sec)
	13	34765	37317	36036	34982	34155	ECT	32963	3.6	0.01	132
	14	30949	32039	31799	31709	30782	ECT	29891	3.0	0.00	125
	15	38261	40328	39464	37637	37218	ECT	36299	2.5	0.01	171
	16	36810	38635	37022	36375	36181	ECT	34506	4.9	0.00	145
	17	33810	36328	35613	36512	34567	STPT	33158	2.0	0.01	151
	18	44473	43577	41993	41478	40174	ECT	39524	1.6	0.00	163
	19	36695	39387	38036	36323	35241	ECT	34686	1.6	0.01	165
	20	36312	39445	38668	36619	36316	STPT	35075	3.5	0.01	147
	21	37532	40009	36950	34869	34660	ECT	33609	3.1	0.00	134
	22	35573	37694	35387	36544	34029	ECT	33084	2.9	0.00	124
	23	37426	40392	39542	37675	38504	STPT	36907	1.4	0.01	123
	24	30919	31672	31195	31685	30167	ECT	28888	4.4	0.01	124
	25	30700	31050	30316	30602	28947	ECT	28675	0.9	0.00	128
	26	35198	37017	35679	34088	34182	SPTL	33173	2.8	0.00	167
	27	36607	38097	36344	36113	35472	ECT	34380	3.2	0.01	168
	28	32896	33422	31832	32316	31084	ECT	30490	1.9	0.00	158
	29	33559	36046	34982	34504	33721	STPT	32597	3.0	0.00	143
	30	31546	33088	32137	30367	30497	SPTL	29682	2.3	0.01	130
5	1	27295	30558	26623	25478	24525	ECT	24205	1.3	0.01	237
	2	27738	25980	24817	25080	23389	ECT	23062	1.4	0.01	244
	3	27301	29618	24375	22970	22260	ECT	21759	2.3	0.01	383
	4	27889	28595	25026	24543	23296	ECT	22718	2.5	0.01	209
	5	26565	27052	25002	24507	23523	ECT	23133	1.7	0.00	341
	6	24983	26682	24171	23313	22167	ECT	22109	0.3	0.00	206
	7	58417	60555	59304	60322	56322	ECT	55386	1.7	0.00	226
	8	63849	67408	66011	62817	61348	ECT	60333	1.7	0.00	316
	9	66005	68380	68893	64572	63534	ECT	61782	2.8	0.00	405
	10	71810	71761	68496	64616	62550	ECT	61820	1.2	0.00	267
	11	63026	63662	62757	59737	58838	ECT	58110	1.3	0.00	646
	12	60083	65399	63820	59517	58901	ECT	57282	2.8	0.00	381
	13	41643	41011	38061	35142	34170	ECT	33395	2.3	0.00	396
	14	33437	34108	32108	30802	29718	ECT	29044	2.3	0.00	450
	15	36269	40501	38017	33643	32975	ECT	32232	2.3	0.00	201
	16	30647	34331	33336	28579	28065	ECT	26879	4.4	0.00	178
	17	33384	38203	37287	33355	33337	ECT	31441	6.0	0.01	540
	18	33107	36818	34196	30509	30706	SPTL	28677	6.4	0.01	188
	19	31019	31883	32765	29562	29254	ECT	28607	2.3	0.01	265
	20	26907	28849	27555	26684	25049	ECT	24379	2.7	0.01	367

(*Continued on next page.*)

Table A3. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_h(sec)	T_{ts}(sec)
	21	41322	48013	43449	39219	38345	ECT	37347	2.7	0.01	281
	22	36981	37516	34818	32402	32214	ECT	30681	5.0	0.01	548
	23	46709	44546	42356	40233	39099	ECT	38058	2.7	0.01	252
	24	30555	31944	30379	27052	26902	ECT	25827	4.2	0.01	496
	25	45284	44881	42602	40364	39044	ECT	38154	2.3	0.01	188
	26	33198	33937	32499	30330	29489	ECT	28682	2.8	0.00	190
	27	38139	40845	37680	34779	34664	ECT	33680	2.9	0.00	189
	28	37912	41316	36999	34132	32591	ECT	31345	4.0	0.01	285
	29	35377	37438	36583	33997	33704	ECT	32626	3.3	0.01	190
	30	37459	40889	39981	36473	35546	ECT	34745	2.3	0.01	330

Table A4. Comparative overview of performance of heuristics for $\sum C_j$: $n = 50; m = 10, 20$

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_h(sec)	T_{ts}(sec)
10	1	16445	16928	14493	15535	13153	ECT	13010	1.1	0.01	225
	2	19802	19869	16307	17863	15007	ECT	14860	1.0	0.01	427
	3	16871	16939	14287	15305	12653	ECT	12533	1.0	0.01	709
	4	15390	16025	13165	14175	12008	ECT	11902	0.9	0.00	201
	5	16489	17031	13844	15395	12743	ECT	12732	0.1	0.01	596
	6	15104	15683	13650	14104	11914	ECT	11811	0.9	0.01	364
	7	65146	69757	66899	63736	61760	ECT	60619	1.9	0.01	1023
	8	69024	72769	70631	67468	65397	ECT	64191	1.9	0.01	458
	9	65593	71470	68701	65997	62926	ECT	61254	2.7	0.01	518
	10	69901	76242	72376	70784	67438	ECT	66041	2.1	0.00	598
	11	68661	71489	70365	67661	64924	ECT	63438	2.3	0.00	869
	12	71462	74963	72780	71077	68116	ECT	67647	0.7	0.00	1078
	13	39470	43744	42157	36566	34935	ECT	34287	1.9	0.00	355
	14	29734	32473	32257	28357	27134	ECT	26469	2.5	0.00	417
	15	34369	36647	34446	32330	30500	ECT	29951	1.8	0.00	566
	16	36813	39809	38865	33681	32356	ECT	31405	2.8	0.00	354
	17	32748	38567	35314	31197	30150	ECT	29053	3.8	0.00	653
	18	31250	37167	33322	29604	28476	ECT	27754	2.6	0.01	270
	19	46213	49030	42724	38334	36593	ECT	36078	1.4	0.01	372
	20	27082	32314	31148	26363	24912	ECT	24745	0.7	0.01	264
	21	30163	37160	34061	28801	27128	ECT	26679	1.7	0.01	683
	22	36538	41095	38867	34356	33205	ECT	32469	2.3	0.01	956
	23	38075	42458	36689	33060	32150	ECT	31393	2.4	0.01	312
	24	34890	35221	34208	33127	31505	ECT	30695	2.6	0.01	441

(*Continued on next page.*)

Table A4. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
	25	36871	39091	40106	34239	33139	ECT	32163	3.0	0.01	294
	26	40009	41748	38859	34735	34582	ECT	33094	4.5	0.01	286
	27	34056	37911	35581	31243	30822	ECT	29468	4.6	0.01	647
	28	28227	31970	30391	27370	25931	ECT	25276	2.6	0.01	303
	29	30336	33376	31835	28940	28414	ECT	27443	3.5	0.00	257
	30	38919	45725	39956	34883	33614	ECT	32650	3.0	0.01	298
20	1	11257	10693	9009	10642	7990	ECT	7959	0.4	0.01	331
	2	11004	10763	8948	10734	8119	ECT	8119	0.0	0.02	289
	3	14123	13352	11018	13334	10054	ECT	10042	0.1	0.01	206
	4	9598	10729	8675	9629	7790	ECT	7790	0.0	0.01	250
	5	10968	12510	8918	10019	8097	ECT	8042	0.7	0.02	372
	6	11777	11676	9569	11361	8632	ECT	8632	0.0	0.02	351
	7	69643	71802	71389	67597	66433	ECT	64816	2.5	0.01	754
	8	68121	72885	71347	69846	65843	ECT	64567	2.0	0.01	1232
	9	70151	71231	71999	69149	66268	ECT	65563	1.1	0.02	857
	10	75644	76240	74021	71605	69380	ECT	68233	1.7	0.01	590
	11	72504	73586	72490	70322	68002	ECT	66595	2.1	0.01	1544
	12	71884	76348	72206	72045	68437	ECT	67589	1.3	0.02	1298
	13	30595	33001	31515	29670	28361	ECT	27704	2.4	0.01	455
	14	35954	43155	39163	33651	34266	SPTL	31818	5.8	0.01	634
	15	34828	37622	34169	31877	31056	ECT	29662	4.7	0.02	727
	16	32524	37727	35799	30473	28754	ECT	27965	2.8	0.02	741
	17	38368	42450	39301	35881	33634	ECT	32781	2.6	0.01	1122
	18	29712	33939	31918	26889	26227	ECT	25006	4.9	0.01	491
	19	30278	32278	31843	29020	28153	ECT	27223	3.4	0.01	953
	20	42371	45842	41950	37667	36945	ECT	35470	4.2	0.02	440
	21	32972	39687	36665	31653	30539	ECT	29438	3.7	0.02	456
	22	31623	39491	34427	29616	29668	SPTL	28307	4.6	0.02	474
	23	31442	38024	35232	30246	28769	ECT	27849	3.3	0.01	1191
	24	36353	44126	41191	35139	34092	ECT	33067	3.1	0.01	435
	25	37727	43079	39031	35718	32971	ECT	31930	3.3	0.01	568
	26	37370	39778	40642	35659	33384	ECT	33115	0.8	0.01	826
	27	39232	43520	40840	36497	34979	ECT	34124	2.5	0.02	448
	28	34981	38952	38915	33878	33577	ECT	32474	3.4	0.02	493
	29	43619	48720	43675	40344	38932	ECT	38080	2.2	0.02	686
	30	33549	36387	36092	32293	30809	ECT	30366	1.5	0.02	604

Table A5. Comparative overview of performance of heuristics for $\sum C_j : n = 100; m = 2, 5$

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_{h}(sec)	T_{ts}(sec)
2	1	201751	213174	207565	203246	200816	ECT	195018	3.0	0.01	1562
	2	205999	213855	210832	209122	206147	STPT	200783	2.6	0.01	1557
	3	204499	205327	203993	205021	199803	ECT	196457	1.7	0.01	1712
	4	191028	194739	194380	195040	190711	ECT	185350	2.9	0.01	1939
	5	182448	190810	189360	192013	184299	STPT	179656	1.6	0.01	1026
	6	180926	183442	182534	181679	177799	ECT	173788	2.3	0.01	2046
	7	216187	222412	221924	213943	209152	ECT	206370	1.3	0.01	1237
	8	195894	201311	200536	199321	196490	STPT	192888	1.6	0.01	1527
	9	200508	206189	205088	201764	200954	STPT	196381	2.1	0.01	1072
	10	221545	231421	221801	214317	209866	ECT	205013	2.4	0.01	1514
	11	206242	220136	216598	209408	208566	STPT	201386	2.4	0.01	1021
	12	213643	223749	223769	217551	216280	STPT	211158	1.2	0.01	1020
	13	157153	165991	158212	149317	147830	ECT	143631	2.9	0.01	1519
	14	166179	173796	163558	160290	157854	ECT	151896	3.9	0.01	1107
	15	174370	180235	171737	165156	164223	ECT	159978	2.7	0.01	1593
	16	157171	156214	152659	154897	149392	ECT	144814	3.2	0.01	1123
	17	153210	162478	156608	153016	153534	SPTL	147269	3.9	0.01	997
	18	128037	136491	134150	132959	129660	STPT	125277	2.2	0.01	978
	19	139517	150046	148639	142968	139635	STPT	135975	2.6	0.01	1018
	20	141024	143590	142133	139021	135478	ECT	132144	2.5	0.01	1104
	21	130790	133171	131435	129222	128839	ECT	125085	3.0	0.00	1113
	22	145814	146745	143823	142601	138710	ECT	136054	2.0	0.01	1158
	23	141637	149349	145730	148807	142014	STPT	136834	3.5	0.01	1920
	24	158325	161330	158544	155472	153393	ECT	148104	3.6	0.01	1116
	25	148989	158511	154313	150759	149599	STPT	144107	3.4	0.01	985
	26	165388	175426	172032	164535	161683	ECT	157099	2.9	0.01	1393
	27	139780	144985	141154	145174	138983	ECT	134754	3.1	0.01	1038
	28	149989	156720	156538	151103	150989	STPT	145229	3.3	0.01	1001
	29	139542	142974	134355	130506	127037	ECT	123435	2.9	0.01	1034
	30	163311	167103	160520	156075	153293	ECT	149279	2.7	0.01	1743
5	1	103994	110052	99651	97515	93032	ECT	91273	1.9	0.02	2577
	2	94761	99806	89461	85230	82581	ECT	80964	2.0	0.01	2172
	3	93280	94335	87444	87790	82232	ECT	81222	1.2	0.02	3498
	4	109957	106437	100354	96899	90966	ECT	90072	1.0	0.01	2087
	5	82072	84702	81806	76945	74836	ECT	73965	1.2	0.01	4243
	6	98101	100491	96893	93273	90403	ECT	88234	2.5	0.02	2979
	7	249554	259712	252916	240689	233617	ECT	228520	2.2	0.01	4605
	8	242514	259491	254702	238001	233039	ECT	227198	2.6	0.01	7113

(*Continued on next page.*)

Table A5. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
	9	250124	250693	246399	246944	234599	ECT	229900	2.0	0.01	4966
	10	245050	249357	248271	241748	234259	ECT	230343	1.7	0.01	2665
	11	243903	254265	250479	243098	233447	ECT	229776	1.6	0.02	2872
	12	239113	249683	251398	238742	228734	ECT	225028	1.6	0.01	6686
	13	151015	156628	153760	141097	137223	ECT	133429	2.8	0.01	1910
	14	128104	139369	134581	120749	117735	ECT	113583	3.7	0.02	4424
	15	118352	121990	117253	110751	106863	ECT	103077	3.7	0.01	2934
	16	114464	125841	119278	114066	108994	ECT	105267	3.5	0.01	2392
	17	122977	128874	125311	116855	114927	ECT	111761	2.8	0.01	3768
	18	133698	151235	148460	131324	129739	ECT	124916	3.9	0.02	2780
	19	141270	145670	141035	131423	129141	ECT	123707	4.4	0.01	1929
	20	112450	123473	113304	105222	100120	ECT	98671	1.5	0.01	2166
	21	122470	126192	117291	108660	107856	ECT	103522	4.2	0.02	3365
	22	134124	143796	139599	120486	121740	SPTL	116780	3.2	0.02	1649
	23	129769	141243	132100	122887	120911	ECT	117557	2.9	0.01	2397
	24	115955	129537	122726	111038	111203	SPTL	105216	5.5	0.01	4970
	25	133180	143945	138104	123345	122986	ECT	118212	4.0	0.02	2979
	26	122546	137444	131057	118425	117392	ECT	113179	3.7	0.02	2302
	27	129092	141098	128155	113314	113220	ECT	108600	4.3	0.01	2230
	28	143415	156544	155445	135081	135905	SPTL	129778	4.1	0.01	2590
	29	131370	142748	135913	123166	123013	ECT	116774	5.3	0.01	2521
	30	123541	140041	134739	119482	117665	ECT	114058	3.2	0.01	2072

Table A6. Comparative overview of performance of heuristics for $\sum C_j : n = 100; m = 10, 20$

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
10	1	51730	51400	44439	48755	40797	ECT	40737	0.1	0.02	2636
	2	58215	62199	53409	52941	47774	ECT	47772	0.0	0.02	2790
	3	75948	72560	54272	57862	49362	ECT	49308	0.1	0.02	3120
	4	61968	64749	51435	57161	48159	ECT	48017	0.3	0.02	2851
	5	60656	67568	56693	55732	49312	ECT	49206	0.2	0.02	7364
	6	54978	54301	44912	52371	42233	ECT	42211	0.1	0.02	3275
	7	276864	282550	275075	263835	252802	ECT	249672	1.3	0.02	4210
	8	257167	270210	263083	254641	244296	ECT	240482	1.6	0.01	6524
	9	259098	268138	264762	257705	244024	ECT	242521	0.6	0.01	3851
	10	260479	275763	265342	253495	245523	ECT	241963	1.5	0.02	4346
	11	259891	269720	268469	257727	247510	ECT	243847	1.5	0.02	8386
	12	267753	283469	277726	265789	256535	ECT	252102	1.8	0.02	13291

(*Continued on next page.*)

Table A6. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
	13	103451	117310	112632	98146	94447	ECT	92113	2.5	0.02	2913
	14	123965	135866	124446	110146	107764	ECT	104830	2.8	0.02	2231
	15	109043	123944	120151	104311	103188	ECT	100297	2.9	0.02	3220
	16	112129	121801	112808	104669	100711	ECT	98938	1.8	0.02	4969
	17	123672	138739	131214	114723	113004	ECT	108668	4.0	0.02	6086
	18	128959	136254	128690	113144	105832	ECT	103901	1.9	0.02	2538
	19	143618	155772	146089	129021	125409	ECT	122594	2.3	0.02	3729
	20	134627	140335	139964	124791	121124	ECT	117117	3.4	0.02	2442
	21	134335	148495	143211	124231	123752	ECT	119466	3.6	0.02	4839
	22	148186	159584	152570	130367	128325	ECT	123734	3.7	0.02	5597
	23	130243	142621	134454	118771	117489	ECT	113190	3.8	0.02	2609
	24	115493	129718	117632	105188	101477	ECT	99313	2.2	0.02	2606
	25	122796	135949	131373	113624	109419	ECT	105929	3.3	0.02	7596
	26	121208	126884	126167	112220	109858	ECT	106125	3.5	0.02	2317
	27	132947	147615	141208	122825	118801	ECT	116182	2.3	0.02	3536
	28	133057	153166	147027	123765	122887	ECT	117968	4.2	0.02	4448
	29	107358	126239	123233	102094	100368	ECT	97256	3.2	0.02	2523
	30	109784	109965	110470	100395	97980	ECT	94984	3.2	0.02	3115
20	1	31507	34709	27525	30990	24747	ECT	24747	0.0	0.03	6391
	2	51461	46619	35878	42608	31491	ECT	31477	0.0	0.03	2847
	3	32125	32806	29098	31060	25387	ECT	25382	0.0	0.03	3132
	4	37878	36863	32590	36348	28273	ECT	28273	0.0	0.03	4022
	5	34968	33774	28264	32763	25172	ECT	25166	0.0	0.03	7097
	6	40499	44293	31757	38119	28112	ECT	28094	0.1	0.03	3972
	7	289001	293160	285506	274469	262438	ECT	260394	0.8	0.03	6526
	8	273900	283547	275002	269019	258144	ECT	256525	0.6	0.03	8440
	9	274384	284745	281724	271487	262668	ECT	258037	1.8	0.03	19790
	10	267783	272543	268307	263080	252054	ECT	248345	1.5	0.03	10674
	11	261362	269259	268743	255660	249175	ECT	245036	1.7	0.03	17883
	12	267170	279653	273371	261601	255429	ECT	252458	1.2	0.02	11975
	13	110099	129673	124382	105393	99857	ECT	97284	2.6	0.03	4876
	14	113829	136348	118819	105227	99864	ECT	98280	1.6	0.03	14629
	15	137730	156302	151811	127682	125522	ECT	122560	2.4	0.03	5754
	16	137083	151313	147545	127781	123397	ECT	120796	2.2	0.03	6812
	17	110439	134767	126446	106747	103577	ECT	100953	2.6	0.03	6907
	18	127908	141958	134976	115439	109811	ECT	108266	1.4	0.03	10421
	19	132345	143943	135258	119921	114191	ECT	110852	3.0	0.02	9334
	20	120176	145148	134570	113565	109159	ECT	107385	1.7	0.03	4868

(*Continued on next page.*)

Table A6. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
	21	116835	125242	126888	108913	104216	ECT	103149	1.0	0.03	5298
	22	119721	135257	128844	110862	104187	ECT	102891	1.3	0.02	7974
	23	127583	143185	137284	115073	110980	ECT	109126	1.7	0.03	5193
	24	120266	135267	134017	116405	110744	ECT	107821	2.7	0.03	19503
	25	107071	122575	115705	99753	94460	ECT	93314	1.2	0.03	5584
	26	130740	147528	150994	123489	120915	ECT	117383	3.0	0.03	4864
	27	132380	151070	144857	124730	121623	ECT	117812	3.2	0.02	5636
	28	133653	139942	134332	123943	115767	ECT	111257	4.1	0.03	8660
	29	127697	149131	127412	112500	107025	ECT	105929	1.0	0.03	7162
	30	127584	155583	148116	121953	116698	ECT	114081	2.3	0.03	10750

Table A7. Comparative overview of performance of heuristics for $\sum C_j : n = 200; m = 2, 5$

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
2	1	751347	783748	779718	777272	763389	STPT	740643	1.4	0.03	9039
	2	829435	854575	853640	828372	815843	ECT	800122	2.0	0.02	10024
	3	768405	782816	778434	783170	768321	ECT	744624	3.2	0.02	14159
	4	795511	808556	811992	816084	796506	STPT	772984	2.9	0.03	16922
	5	794570	822327	815773	815454	805579	STPT	784393	1.3	0.02	10488
	6	842382	865672	856989	842777	830685	ECT	807824	2.8	0.03	20416
	7	843822	856501	848258	836643	828090	ECT	808598	2.4	0.02	14519
	8	837933	858716	840240	833960	821696	ECT	802098	2.4	0.03	9623
	9	818484	830768	831279	817138	801141	ECT	785153	2.0	0.03	9341
	10	821884	849591	846073	835875	831064	STPT	807313	1.8	0.03	13372
	11	825740	845863	832096	827147	811036	ECT	792398	2.4	0.02	18337
	12	816934	852053	842047	829034	827049	STPT	798687	2.3	0.03	8550
	13	524535	549043	540464	537407	532650	STPT	512362	2.4	0.03	7778
	14	597476	621466	610716	603040	593563	ECT	569578	4.2	0.02	10304
	15	569398	599351	591598	593068	580237	STPT	555122	2.6	0.03	8155
	16	576178	598529	582215	585182	569821	ECT	550376	3.5	0.03	8235
	17	581322	617125	606928	590374	586522	STPT	560592	3.7	0.03	8172
	18	631182	632364	599318	574009	570848	ECT	559548	2.0	0.02	10899
	19	540160	561198	544520	537284	531893	ECT	512133	3.9	0.03	9236
	20	536433	556059	542984	542576	532878	ECT	513987	3.7	0.03	8019
	21	465829	485368	481019	482458	469719	STPT	456290	2.1	0.03	7833
	22	564995	591358	576411	567375	565601	STPT	542035	4.2	0.02	8608
	23	536922	571798	550490	536515	529926	ECT	508640	4.2	0.03	9401

(*Continued on next page.*)

Table A7. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_{h}(sec)	T_{ts}(sec)
	24	549608	588418	561030	543699	529878	ECT	512792	3.3	0.03	9611
	25	580357	611249	594751	581161	565406	ECT	550480	2.7	0.03	8224
	26	591139	617955	605809	595336	586403	ECT	564890	3.8	0.02	8247
	27	527452	547236	530931	546358	518392	ECT	499358	3.8	0.02	8236
	28	524515	534706	528454	534962	515166	ECT	503053	2.4	0.02	9529
	29	581755	617815	603057	578097	584974	SPTL	558522	3.5	0.02	9393
	30	571783	607625	592904	575297	564737	ECT	540480	4.5	0.02	8378
5	1	371409	387267	364436	347015	340158	ECT	333817	1.9	0.04	19231
	2	364931	394215	349997	334870	330505	ECT	323557	2.1	0.04	13158
	3	380434	390428	360494	344328	338138	ECT	333884	1.3	0.04	17198
	4	384166	388334	364814	352035	344302	ECT	337819	1.9	0.04	11451
	5	406444	420561	379426	362291	348197	ECT	339536	2.6	0.03	19162
	6	399254	412515	370736	362558	345531	ECT	339335	1.8	0.03	10615
	7	979165	1011189	1010951	984742	939740	ECT	924589	1.6	0.03	16872
	8	931835	993234	980901	926650	893182	ECT	873905	2.2	0.03	25568
	9	987612	1010520	988983	950505	918844	ECT	896552	2.5	0.03	37098
	10	939733	977652	957589	943436	904757	ECT	889579	1.7	0.04	29036
	11	938173	1004099	994097	953533	930099	ECT	908780	2.3	0.03	14904
	12	991293	1012317	993323	950901	920978	ECT	907160	1.5	0.03	25014
	13	486650	521557	490995	441174	441986	SPTL	421258	4.7	0.04	9633
	14	454887	505987	486880	434019	434275	SPTL	413464	5.0	0.04	8742
	15	516277	557034	536091	481149	482783	SPTL	461344	4.3	0.03	13554
	16	417998	463566	448585	402387	402663	SPTL	384162	4.7	0.03	9900
	17	491101	543844	521061	457016	462205	SPTL	440006	3.9	0.03	12801
	18	556129	621752	589021	521635	524978	SPTL	497944	4.8	0.03	12951
	19	439837	473619	467456	409579	413778	SPTL	390019	5.0	0.04	9100
	20	394058	448084	443564	387700	391943	SPTL	373971	3.7	0.04	13760
	21	440190	485695	474575	418905	421295	SPTL	403212	3.9	0.04	12710
	22	523662	566825	543375	492751	484907	ECT	471785	2.8	0.04	13153
	23	463979	503138	482871	440062	438458	ECT	419693	4.5	0.04	18271
	24	488739	542436	519433	469309	462833	ECT	445767	3.8	0.04	13992
	25	458310	521344	494502	434739	444621	SPTL	418990	3.8	0.03	16491
	26	473525	541779	515238	445493	454376	SPTL	431600	3.2	0.03	12957
	27	428752	487710	466674	412079	416598	SPTL	400603	2.9	0.03	10831
	28	540924	581511	536442	482758	479520	ECT	464309	3.3	0.03	12109
	29	428606	495622	467657	418544	422376	SPTL	404091	3.6	0.04	12810
	30	455252	514794	481892	431880	438531	SPTL	415521	3.9	0.04	16107

Table A8. Comparative overview of performance of heuristics for $\sum C_j : n = 200; m = 10, 20$

m	#	STPT	SMPT	SMCT	SPTL	ECT	$H_{\min}$	TS	Imp	T_h(sec)	T_{ts}(sec)
10	1	237466	252118	193219	194552	177337	ECT	177115	0.1	0.05	11201
	2	209844	210412	189056	196525	171373	ECT	171286	0.1	0.05	14426
	3	199665	203585	181484	185055	164176	ECT	163615	0.3	0.05	27724
	4	187374	191601	178248	176727	160612	ECT	160612	0.0	0.05	17901
	5	205714	202787	183809	188188	167676	ECT	167676	0.0	0.05	20128
	6	235584	232684	198005	204483	182980	ECT	182980	0.0	0.05	24718
	7	1001510	1054578	1044552	999337	969660	ECT	952735	1.8	0.04	38809
	8	1004234	1060654	1035245	1004700	965507	ECT	951801	1.4	0.05	53767
	9	1009360	1065229	1052366	989070	965537	ECT	950044	1.6	0.05	32679
	10	1028652	1092001	1069718	1012878	986717	ECT	975016	1.2	0.04	37607
	11	1013691	1059720	1036856	1013642	967850	ECT	955816	1.3	0.04	53849
	12	966551	1043540	1025245	971817	939464	ECT	926601	1.4	0.05	25654
	13	432196	514772	486009	409766	405781	ECT	396874	2.2	0.05	15840
	14	445836	511796	499376	422411	414488	ECT	401328	3.3	0.05	31829
	15	501818	563292	545586	466918	462835	ECT	447521	3.4	0.05	20877
	16	496843	556241	514362	449573	444749	ECT	435310	2.2	0.05	23783
	17	443639	505098	496368	420812	419896	ECT	403999	3.9	0.05	33041
	18	424354	485555	468443	403186	404185	SPTL	387939	3.9	0.05	30047
	19	462215	525206	505369	442377	436809	ECT	426006	2.5	0.05	20468
	20	430282	498522	467904	403306	396221	ECT	384520	3.0	0.04	33976
	21	430870	469812	447524	403985	390384	ECT	383033	1.9	0.05	17944
	22	409306	467463	458183	385421	378707	ECT	371715	1.9	0.05	18682
	23	457875	508879	482532	426599	409242	ECT	399485	2.4	0.05	21013
	24	460078	489723	481575	418095	406688	ECT	396200	2.6	0.05	26122
	25	425734	488920	467073	401898	393922	ECT	384122	2.6	0.05	27370
	26	467291	520761	490647	427564	421776	ECT	410260	2.8	0.05	19501
	27	444648	505667	478840	416005	408335	ECT	401643	1.7	0.04	17608
	28	442567	506098	489526	411344	410833	ECT	397857	3.3	0.05	26183
	29	514729	592078	564618	487345	482476	ECT	471690	2.3	0.05	17268
	30	428779	500557	481875	414425	411424	ECT	396966	3.6	0.05	25527
20	1	124477	125115	103892	117520	91383	ECT	91383	0.0	0.08	19609
	2	124245	120029	104187	114153	90937	ECT	90937	0.0	0.08	10598
	3	139246	138588	117410	133359	101366	ECT	101366	0.0	0.08	18463
	4	142961	137936	114813	130181	99859	ECT	99859	0.0	0.08	23749
	5	117184	115943	98580	105452	87375	ECT	87375	0.0	0.08	10709
	6	118391	126743	103768	114724	90801	ECT	90723	0.1	0.31	71259
	7	1043204	1112618	1091234	1027525	994221	ECT	984208	1.0	0.07	58318

(*Continued on next page.*)

Table A8. (*Continued*).

m	#	STPT	SMPT	SMCT	SPTL	ECT	H_{min}	TS	Imp	T_h(sec)	T_{ts}(sec)
	8	1050091	1079484	1074355	1039653	1002311	ECT	994160	0.8	0.07	57028
	9	1018101	1064884	1055257	1020095	983185	ECT	974861	0.9	0.07	74136
	10	1054917	1097954	1061766	1030505	988659	ECT	985036	0.4	0.07	33636
	11	1009557	1082353	1057005	1007310	980984	ECT	973127	0.8	0.07	67100
	12	1052401	1116565	1088463	1035568	1012069	ECT	1001611	1.0	0.07	59941
	13	418231	495028	488889	400110	393625	ECT	384994	2.2	0.07	29548
	14	472314	515464	491113	436585	423977	ECT	415846	2.0	0.07	24688
	15	536724	623068	566308	489435	470414	ECT	464730	1.2	0.07	36949
	16	450619	511593	494448	431282	418664	ECT	408352	2.5	0.07	62997
	17	424093	527320	496181	406218	393643	ECT	387013	1.7	0.07	61406
	18	462096	506016	493850	423612	407513	ECT	402501	1.2	0.08	39983
	19	434712	520330	491568	419980	409574	ECT	402754	1.7	0.08	53691
	20	420095	498411	475054	401200	388663	ECT	382046	1.7	0.08	32616
	21	423987	488693	469739	400226	384071	ECT	377221	1.8	0.08	30898
	22	443209	522240	504893	424467	415818	ECT	408354	1.8	0.08	43604
	23	440845	503560	482871	420220	407102	ECT	398894	2.1	0.08	41930
	24	469837	535533	518640	437886	419401	ECT	408859	2.6	0.08	58420
	25	417422	506891	470564	403772	388220	ECT	382673	1.4	0.07	71862
	26	503378	537893	527312	461392	444382	ECT	435438	2.1	0.07	32846
	27	446719	523702	500742	422720	406579	ECT	400297	1.6	0.07	27337
	28	496992	522920	503781	442683	432434	ECT	425862	1.5	0.07	26981
	29	417753	505517	481878	400137	389781	ECT	382463	1.9	0.07	43395
	30	477827	529550	496833	429560	414017	ECT	407827	1.5	0.07	53948

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by the National Science Foundation through grant DMI 0300156. Computing facilities are provided by the PC Cluster Lab of the CS Department at NJIT, which is partially supported by NSF grant NSF-9977508 and partially by NJIT SBR grant SBR-421350. The authors are also grateful to the referees for their many helpful comments.

REFERENCES

Barnes, J., M. Laguna, and F. Glover, "An overview of Tabu Search approaches to production scheduling problems," in D. Brown and W. Scherer (eds.), *Intelligent Scheduling Systems*, Kluwer Academic Publishers, 1995, pp. 101–127.

Du, J. and J. Y.-T. Leung, "Minimizing total tardiness on one machine is NP-hard," *Mathematics of Operations Research*, **15**, 483–495 (1990).

Emmons, H., "One-machine sequencing to minimize certain functions of job tardiness," *Operations Research*, **17**, 701–715 (1969).

Garey, M. R. and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

Glover, F. and M. Laguna, *Tabu Search*, Kluwer Academic Publishers, Boston, USA, 1997.

Graham, R. L., E. L. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan, "Optimization and approximation in deterministic sequencing and scheduling: A survey," *Annals of Discrete Mathematics*, **5**, 287–326 (1979).

Julien, F. M. and M. J. Magazine, "Scheduling customer orders—An alternative production scheduling approach," *Journal of Manufacturing and Operations Management*, **3**, 177–199 (1990).

Laguna, M., "Implementing and testing the Tabu Cycle and conditional probability methods, Working paper, 2004.

Lenstra, J. K., "Sequencing by enumerative methods," Mathematical Centre Tracts 69, Mathematisch Centrum, Amsterdam, The Netherlands, 1977.

Leung, J. Y-T., H. Li, M. Pinedo, and S. Sriskandarajah, "Open shops with jobs overlap—revisited," *European Journal of Operational Research*, **163**(2), 569–571 (2005).

Li, H., "Order scheduling in dedicated and flexible machine environment," PhD Thesis, Department of Computer Science, New Jersey Institute of Technology, Newark, New Jersey, 2005.

Smith, W. E., "Various optimizers for single stage production," *Naval Research Logistics Quarterly*, **3**, 59–66 (1956).

Sung, C. S. and S. H. Yoon, "Minimizing total weighted completion time at a pre-assembly stage composed of two feeding machines," *International Journal of Production Economics*, **54**, 247–255 (1998).

Wagneur, E. and C. Sriskandarajah, "Open shops with jobs overlap," *European Journal of Operational Research*, **71**, 366–378 (1993).

Wang, G. and T. C. E. Cheng, "Customer order scheduling to minimize total weighted completion time," in *Proceedings of the 1st Multidisciplinary Conference on Scheduling Theory and Applications*, 2003, pp. 409–416.

Yang, J., "Scheduling with batch objectives. PhD Thesis, Industrial and Systems Engineering Graduate Program, The Ohio State University, Columbus, Ohio, 1998.

Zweben, M. and M. Fox (eds.), *Intelligent Scheduling*, Morgan Kaufmann Publishers, 1994.