Centrum voor Wiskunde en Informatica
Centre for Mathematics and Computer Science

J.A. Hoogeveen, S.L. van de Velde

Polynomial-time algorithms for single-machine multicriteria scheduling

Department of Operations Research, Statistics, and System Theory Report BS-R9008 May

The Centre for Mathematics and Computer Science is a research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a nonprofit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands Organization for the Advancement of Research (N.W.O.).

Polynomial-Time Algorithms for Single-Machine Multicriteria Scheduling

J.A. Hoogeveen
S.L. van de Velde
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam
The Netherlands

We address the problem of scheduling n independent jobs on a single machine so as to minimize multiple criteria. We consider three types of problems. The first one involves the minimization of an arbitrary nondecreasing function of total completion time and an arbitrary nondecreasing minmax cost function. We present an $O(n^3 \min\{(n, \log n + \log p_{max}\})$ time algorithm, where p_{max} is the maximum job processing time. The algorithm can be improved to run in $O(n^3)$ time for the special case that the second objective is the maximum lateness. The second problem is to minimize a nondecreasing linear function of total completion time and maximum earliness. We prove that this problem is solvable in $O(n^4)$ time if the total completion time outweighs the maximum earliness. The third problem involves the minimization of a nondecreasing linear function of maximum earliness and maximum lateness, where preemption is allowed. We present an $O(n \log n)$ time algorithm for this problem.

(1980) Mathematics Subject Classification (Revision 1985): 90B35.
Key Words & Phrases: single-machine scheduling, multicriteria scheduling, composite objective function, Pareto optimality, Pareto-optimal set, extreme set, total completion time, maximum cost, maximum lateness, maximum earliness.
Note: This paper has been submitted for publication.

1. Introduction

A single-machine job shop can be described as follows. A set of n independent jobs has to be scheduled on a single machine that is continuously available and that can process at most one job at a time. Each job J_i ($i = 1, \ldots, n$) requires an uninterrupted positive processing time p_i and has a due date d_i. Without loss of generality, we may assume that the processing times and due dates are integral. A *schedule* σ defines for each job J_i its completion time C_i such that the jobs do not overlap in their execution. A *performance measure* or *scheduling criterion* associates a value $f(\sigma)$ with each feasible schedule σ. Well-known measures are total completion time ΣC_i, maximum lateness L_{max}, defined as $\max_{1 \leqslant i \leqslant n}(C_i - d_i)$, and maximum earliness E_{max}, defined as $\max_{1 \leqslant i \leqslant n}(d_i - C_i)$. In addition, we define γ_{max} as $\gamma_{max} = \max_{1 \leqslant i \leqslant n} \gamma_i(C_i)$, where γ_i is an arbitrary regular cost function for J_i, $i = 1, \ldots, n$. A performance measure is *regular* if it is nondecreasing in the job completion times; total completion time and maximum lateness are of this type. A schedule σ^* is *optimal* for a given performance measure if $f(\sigma^*) = \min_{\sigma \in \Omega} f(\sigma)$, where Ω denotes the set of feasible schedules. Note that in case of a regular performance measure, there is an optimal schedule such that no job

Report BS-R9008
Centre for Mathematics and Computer Science
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

can start earlier without affecting the start time of any other job. This implies that we can restrict ourselves to schedules that do not contain idle time. Therefore, a sequence or permutation of the n jobs defines a unique schedule.

Since the beginning of machine scheduling research more than thirty years ago, most research has been concerned with single performance measures. Recently, the notion gains ground that real life scheduling has to take several performance measures into account. Basically, there are two methods to cope with multiple criteria. If the objectives are subject to a hierarchy, the objectives are considered *sequentially* in order of relevance. An example hereof is the problem of minimizing maximum tardiness subject to the minimum number of tardy jobs (Shanthikumar, 1983); the primary criterion is to minimize the number of tardy jobs, and subject to this, the maximum tardiness is minimized.

This paper, however, is concerned with the *simultaneous* optimization of several criteria. In this alternative approach, the performance measures, specified by the functions f_k $(k = 1, \dots, K)$, are transformed into one single *composite objective* function $F: \Omega \to \mathbb{R}$. With each schedule σ we associate a point $(f_1(\sigma), \dots, f_K(\sigma))$ in $\mathbb{R}^K$ and a value $F(f_1(\sigma), \dots, f_K(\sigma))$. In the remainder, the terms schedule and point are used interchangeably. The associated problem, from now on referred to as problem (P), is formulated as

$$\min_{\sigma \in \Omega} F(f_1(\sigma), \dots, f_K(\sigma)), \tag{P}$$

where F is nondecreasing in each of its arguments. Minimizing the number of tardy jobs and maximum tardiness simultaneously (Nelson et al., 1986) is an example of this method.

A natural question is whether problem (P) is solvable in polynomial time for a given function F. In fact, we can solve this problem in polynomial time for any function F that is nondecreasing in its arguments if we can identify all of the so-called *Pareto-optimal* schedules in polynomial time.

DEFINITION 1. A schedule $\sigma \in \Omega$ is *Pareto-optimal* with respect to the objective functions $f_1, \dots, f_K$ if there is no schedule $\pi \in \Omega$ such that $f_k(\pi) \leqslant f_k(\sigma)$ for all $k = 1, \dots, K$, and $f_k(\pi) < f_k(\sigma)$ for at least one k, $k = 1, \dots, K$ (cf. Figure 1).

THEOREM 1. *Let $F: \sigma \to F(f_1(\sigma), \dots, f_K(\sigma))$ be a composite objective function that is nondecreasing in each argument f_k for $k = 1, \dots, K$. Then there is a Pareto-optimal schedule with respect to the performance criteria $f_1, \dots, f_K$ that solves problem* (P).

Once the Pareto-optimal set (i.e., the set of all Pareto-optimal schedules with respect to the functions $(f_1, \dots, f_K)$ has been determined, problem (P) can be solved for any function F that is nondecreasing in each of its arguments. As a consequence, if each Pareto-optimal schedule can be found in polynomial time and if the cardinality of the Pareto-optimal set is bounded by a polynomial in n, then problem (P) is polynomially solvable.

An interesting subclass of (P) is one in which the composite objective function is *linear*. The associated problem, hereafter referred to as problem (P_α), is formulated as

$$\min_{\sigma \in \Omega} F_\alpha(\sigma) = \min_{\sigma \in \Omega} \textstyle\sum_{k=1}^{K} \alpha_k f_k(\sigma), \tag{P_α}$$

where $\alpha = (\alpha_1, \dots, \alpha_K)$ is a given vector of real nonnegative weights. In analogy to problem

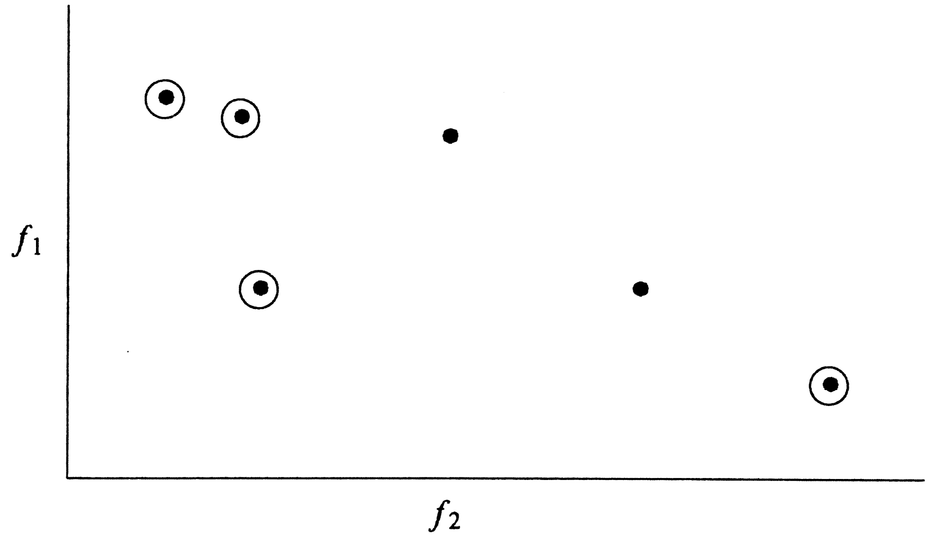

FIGURE 1. The set of Pareto-optimal points.

(P), we are interested in the set of schedules that contains an optimal solution to problem (P_α) for *any* weight vector $\alpha \geqslant 0$. We may restrict ourselves to a subset of the Pareto-optimal set, which we define as the set of *extreme* schedules.

DEFINITION 2. A schedule $\sigma \in \Omega$ is *efficient* with respect to the objective functions $f_1, \ldots, f_K$ if there exists a real vector $\alpha = (\alpha_1, \ldots, \alpha_K) \geqslant 0$ such that $F_\alpha(\sigma) \leqslant F_\alpha(\pi)$ for all schedules $\pi \in \Omega$.

DEFINITION 3. The *efficient frontier* is the shortest curve that connects all efficient points (cf. Figure 2).

DEFINITION 4. A schedule $\sigma \in \Omega$ is *extreme* with respect to the objective functions $f_1, \ldots, f_K$ if it corresponds to a vertex of the efficient frontier.

THEOREM 2. *Let* $F_\alpha: \sigma \rightarrow \Sigma_{k=1}^K \alpha_k f_k(\sigma)$ *be a linear composite objective function, where all weights* $\alpha_1, \ldots, \alpha_K$ *are nonnegative. Then there is an extreme schedule with respect to the performance criteria* $f_1, \ldots, f_K$ *that solves problem* (P_α).

Once the set of extreme schedules with respect to the objective functions $f_1, \ldots, f_K$ has been identified, problem (P_α) can be solved for any given $\alpha \geqslant 0$.

Throughout the paper, we adopt and extend the notation of Graham et al. (1979) to classify scheduling problems with multiple criteria. For instance, $1 \mid \mid F(\Sigma C_i, L_{\max})$ denotes the problem of minimizing an arbitrary nondecreasing function of total completion time and maximum lateness on a single machine, while $1 \mid \mid \alpha_1 \Sigma C_i + \alpha_2 L_{\max}$ denotes its linear counterpart.

In Section 2 we present some fundamental algorithms and notation. In Section 3 we consider

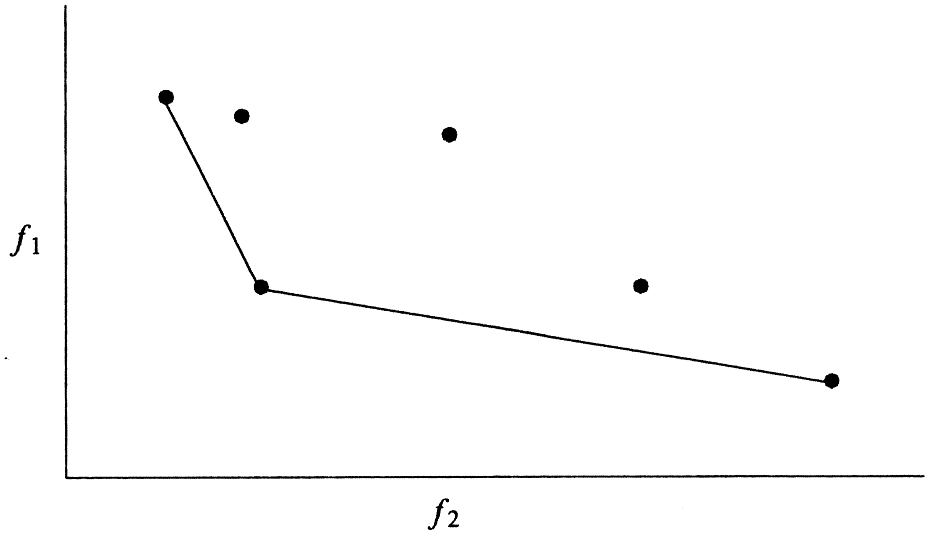

FIGURE 2. The efficient frontier.

the general $1 \mid \mid F(\Sigma C_i, \gamma_{max})$ problem. We establish that Van Wassenhove and Gelders' conjecturedly pseudo-polynomial algorithm (Van Wassenhove and Gelders, 1980) is in fact polynomial: $1 \mid \mid F(\Sigma C_i, \gamma_{max})$ is solvable in $\min\{O(n^4), O(n^3(\log n + \log p_{max}))\}$ time, where $p_{max} = \max_i p_i$, and $1 \mid \mid F(\Sigma C_i, L_{max})$ is solvable in $O(n^3)$ time. These results make the branch-and-bound algorithms proposed by Sen and Gupta (1983) and Nelson et al. (1986) obsolete.

In Section 4, we consider $1 \mid pmtn \mid F(\Sigma C_i, E_{max})$; the notation *pmtn* signifies that job splitting is allowed, that is, the execution of a job can be interrupted and resumed later. The main results are that $1 \mid nmit, pmtn \mid \alpha_1 \Sigma C_i + \alpha_2 E_{max}$ and $1 \mid nmit \mid \alpha_1 \Sigma C_i + \alpha_2 E_{max}$, the latter if $\alpha_1 \geqslant \alpha_2$, are solvable in $O(n^4)$ time, where the notation *nmit* denotes that no machine idle time is allowed.

Gupta and Sen (1984) and Tegze and Vlach (1988) present branch-and-bound algorithms for $1 \mid \mid \alpha_1 L_{max} + \alpha_2 E_{max}$. Hoogeveen (1990) shows that $1 \mid \mid F(L_{max}, E_{max})$ is solvable in $O(n^2 \log n)$ time, if machine idle time is forbidden or if $F(L_{max}, E_{max})$ is linear. In Section 5, we consider the preemptive problem $1 \mid pmtn \mid \alpha_1 L_{max} + \alpha_2 E_{max}$ and prove that it is solvable in $O(n \log n)$ time.

In a subsequent paper (Hoogeveen and Van de Velde, 1990), we show that the algorithms for $1 \mid \mid \alpha_1 \Sigma C_i + \alpha_2 L_{max}$, $1 \mid \mid \alpha_1 \Sigma C_i + \alpha_2 E_{max}$, and $1 \mid pmtn \mid \alpha_1 L_{max} + \alpha_2 E_{max}$ can be applied to find a lower bound for $1 \mid \mid \Sigma C_i + L_{max} + E_{max}$ that dominates than the one proposed by Sen et al. (1988).

We start by stating a few basic algorithms for single-machine single-criterion scheduling problems and introducing some notation.

2. FUNDAMENTAL ALGORITHMS AND NOTATION

There are four single-machine single-criterion scheduling problems related to the bicriteria problems we consider. These involve the minimization of ΣC_i, $L_{\max}$, $E_{\max}$, and $\gamma_{\max}$, respectively. The first three problems are solvable by arranging the jobs in a certain *priority order*, which can be specified in terms of the parameters of the problem type.

THEOREM 3 (Smith, 1956). *ΣC_i is minimized by sequencing the jobs according to the shortest-processing-time (SPT) rule, that is, in order of nondecreasing p_i.*

THEOREM 4 (Jackson, 1955). *$L_{\max}$ is minimized by sequencing the jobs according to the earliest-due-date (EDD) rule, that is, in order of nondecreasing d_i.*

THEOREM 5. *$E_{\max}$ subject to no machine idle time is minimized by sequencing the jobs according to the minimum-slack-time (MST) rule, that is, in order of nondecreasing $d_i - p_i$.*

The fundamental argument that validates each algorithm is the following. Suppose that there is an optimal schedule with two adjacent jobs that are not scheduled according to the indicated priority order. The interchange of the jobs will possibly improve, but certainly not worsen the objective value. An improvement contradicts the claimed optimality, and in the other case, by repetition of the argument, we obtain a schedule with equal objective value that matches the priority order.

THEOREM 6 (Lawler, 1973). *$\gamma_{\max}$ is minimized as follows: while there are unassigned jobs, assign the job that has minimum cost when scheduled in the last unassigned position in that position.*

The optimal objective values for these single-machine scheduling problems will be referred to as ΣC_i^*, $L_{\max}^*$, $E_{\max}^*$, and $\gamma_{\max}^*$, respectively. Furthermore, $\Sigma C_i(\sigma)$, $L_{\max}(\sigma)$, $E_{\max}(\sigma)$, and $\gamma_{\max}(\sigma)$ denote the values of the performance measures in the schedule σ. In analogy, $C_i(\sigma)$, $L_i(\sigma)$, $E_i(\sigma)$, and $\gamma_i(\sigma)$ denote the respective measures for J_i $(i = 1, \ldots, n)$. Whenever (σ) is omitted, we are considering the performance measure in a generic sense, or there is no confusion possible as to the schedule we are referring to. The schedules that minimize ΣC_i, $L_{\max}$, and $E_{\max}$ are referred to as *SPT, EDD,* and *MST*, respectively.

3. MINIMIZING TOTAL COMPLETION TIME AND MAXIMUM COST

Let $\gamma_i : \Omega \rightarrow \mathbb{R}$ denote a regular cost function for job J_i, $i = 1, \ldots, n$, and let $\gamma_i(C_i)$ accordingly denote the cost incurred if job J_i is completed at time C_i. In addition, let $\gamma_{\max} = \max_i \gamma_i(C_i)$. We prove that the $1 \| F(\Sigma C_i, \gamma_{\max})$ problem is solvable in $\min\{O(n^4), O(n^3(\log n + \log p_{\max}))\}$ time, with $p_{\max} = \max_i p_i$, for any function F that is nondecreasing in both ΣC_i and $\gamma_{\max}$. Note that $1 \| F(\Sigma C_i, L_{\max})$ corresponds to a special case of $1 \| F(\Sigma C_i, \gamma_{\max})$.

In Theorem 6, we recalled Lawler's $O(n^2)$ time algorithm for the $1 \| \gamma_{\max}$ problem. An extension is provided by Emmons (1975), who considered the hierarchical problem of minimizing ΣC_i subject to minimum maximum cost $\gamma_{\max}^*$, that is, the $1 | \gamma_{\max} \leqslant \gamma_{\max}^* | \Sigma C_i$ problem. Once $\gamma_{\max}^*$ has been determined by Lawler's algorithm, Emmons' algorithm requires $O(n^2)$ time to minimize total completion time subject to minimum maximum cost. Observe, however, that an upper bound on $\gamma_i(C_i)$ induces a deadline $\bar{d}_i$ on the completion of J_i. Each deadline can be determined in $O(\log(\Sigma p_i))$ time by binary search over the $O(\Sigma p_i)$ possible completion times. Note that $\bar{d}_i$ is computed in constant time if γ_i has an inverse. Once the deadlines have been computed, the problem in the second phase is to minimize total completion time subject to deadlines, $1 \mid \bar{d}_i \mid \Sigma C_i$, which requires only $O(n \log n)$ time (Smith, 1956).

We state the algorithm for $1 \mid \gamma_{\max} \leqslant \gamma \mid \Sigma C_i$, where γ is some upper bound on the cost of the schedule.

ALGORITHM I (Smith, 1956)

Step 1. $T \leftarrow \Sigma p_i$; $J \leftarrow \{J_1, \ldots, J_n\}$.

Step 2. Compute for each job J_i the deadline $\bar{d}_i$ induced by $\gamma_i(C_i) \leqslant \gamma$.

Step 3. Determine $U \leftarrow \{J_i \in J \mid \bar{d}_i \leqslant T\}$, which is the set of jobs that are allowed to be completed at time T.

Step 4. Let J_j be such that $p_j = \max_{J_i \in U} p_i$; in case of ties, choose J_j with the least $\gamma_j(T)$.

Step 5. $J \leftarrow J \setminus \{J_j\}$; $T \leftarrow T - p_j$; If $T > 0$, go to Step 3, else stop.

THEOREM 7. *Algorithm I determines a Pareto-optimal point with respect to ΣC_i and $\gamma_{\max}$.*

PROOF. It suffices to show that the algorithm generates a schedule σ that solves the $1 \mid \gamma_{\max} \leqslant \gamma \mid \Sigma C_i$ and the $1 \mid \Sigma C_i \leqslant \Sigma C_i(\sigma) \mid \gamma_{\max}$ problem simultaneously. Evidently, σ solves $1 \mid \gamma_{\max} \leqslant \gamma \mid \Sigma C_i$. Assume that not σ, but π is optimal for $1 \mid \Sigma C_i \leqslant \Sigma C_i(\sigma) \mid \gamma_{\max}$. This implies $\gamma_{\max}(\pi) < \gamma_{\max}(\sigma) \leqslant \gamma$, and hence, π is also feasible for $1 \mid \gamma_{\max} \leqslant \gamma \mid \Sigma C_i$. Therefore, we have $\Sigma C_i(\pi) = \Sigma C_i(\sigma)$. Compare the two schedules, starting at the end. Suppose the first difference occurs at the kth position, which is occupied by jobs J_i and J_j in σ and π, respectively. Since $\gamma_{\max}(\pi) < \gamma$ and because of the choice of job J_i in the algorithm, we have $p_i \geqslant p_j$. If $p_i > p_j$, then π cannot be optimal: the interchange of these jobs in π, which is feasible, would decrease the total completion time. Hence, it must be that $p_i = p_j$, and because of the choice of job J_i in the algorithm, $\gamma_i(C_i(\sigma)) \leqslant \gamma_j(C_j(\pi))$. This means, however, that the jobs J_i and J_j can be interchanged in the schedule π without affecting the cost of the schedule. Repetition of this argument shows that π can be transformed into σ without affecting the cost, thereby contradicting the assumption that $\gamma_{\max}(\pi) < \gamma_{\max}(\sigma)$. Therefore, the schedule σ also solves the $1 \mid \Sigma C_i \leqslant \Sigma C_i(\sigma) \mid \gamma_{\max}$ problem. Hence, σ is Pareto-optimal with respect to ΣC_i and $\gamma_{\max}$. □

It is obvious that the maximum cost of each Pareto-optimal schedule ranges from $\gamma^*_{\max}$ to $\gamma_{\max}(SPT)$, where for the SPT-order ties are settled in order to minimize maximum cost. The next algorithm, which is similar to Van Wassenhove and Gelders' algorithm, exploits this property for finding the Pareto-optimal set.

ALGORITHM II

Step 1. Compute $\gamma^*_{\max}$ and $\gamma_{\max}(SPT)$; Let $k \leftarrow 1$.

Step 2. Solve the $1 \mid \gamma_{\max} \leqslant \gamma_{\max}(SPT) \mid \Sigma C_i$ problem. This produces the first Pareto-optimal schedule $\sigma^{(1)}$ and Pareto-optimal point $(\Sigma C_i(\sigma^{(1)}), \gamma_{\max}(\sigma^{(1)}))$.

Step 3. If $\gamma_{\max}(\sigma^{(k)}) = \gamma^*_{\max}$, stop. Else $k \leftarrow k + 1$;

Step 4. Solve $1 \mid \gamma_{\max} < \gamma_{\max}(\sigma^{(k-1)}) \mid \Sigma C_i$. This produces the kth Pareto-optimal schedule $\sigma^{(k)}$ and Pareto-optimal point $(\Sigma C_i(\sigma^{(k)}), \gamma_{\max}(\sigma^{(k)}))$. Go to step 3.

THEOREM 8. *Algorithm II determines all Pareto-optimal points with respect to* ΣC_i *and* $\gamma_{\max}$.

PROOF. The proof follows immediately from Theorem 7 and the observation that the cost of each optimal schedule ranges from $\gamma^*_{\max}$ to $\gamma_{\max}(SPT)$. □

A crucial issue is the number of Pareto-optimal points generated by Algorithm II. In the remainder of this section, we prove that there are $O(n^2)$ such schedules, thereby establishing the polynomial nature of the algorithm.

Let $S_i(\sigma)$ denote the start time of job J_i in schedule σ. We define

$$\delta_{ij}(\sigma) = \begin{cases} 1 & \text{if } S_i(\sigma) < S_j(\sigma) \text{ and } p_i > p_j, \\ 0 & \text{otherwise,} \end{cases}$$

and $\Delta(\sigma) = \Sigma_{i,j}\, \delta_{ij}(\sigma)$. Note that if $\delta_{ij}(\sigma) = 1$, the interchange of the jobs J_i and J_j will decrease the total completion time. In that respect, $\delta_{ij}(\sigma) = 1$ signals a *positive* interchange. Observe that $\Delta(SPT) = 0$ and $\Delta(\sigma) \leqslant \frac{1}{2}n(n-1)$ for any $\sigma \in \Omega$. In addition, we define a *neutral* interchange with respect to σ as the interchange of two jobs J_i and J_j with $p_i = p_j$.

LEMMA 1. *If the schedule* π *can be obtained from schedule* σ *through one positive interchange, then* $\Delta(\pi) < \Delta(\sigma)$.

PROOF. Suppose that J_i and J_j, with $p_i > p_j$, are the jobs that have been interchanged. The interchange affects only the jobs scheduled between J_i and J_j. For some job J_l with $S_i(\sigma) < S_l(\sigma) < S_j(\sigma)$ it is easy to verify that $\delta_{il}(\sigma) + \delta_{lj}(\sigma) \geqslant \delta_{jl}(\pi) + \delta_{li}(\pi)$. □

THEOREM 9. *Consider two Pareto-optimal schedules* σ *and* π. *If* $\Sigma C_i(\sigma) < \Sigma C_i(\pi)$ *then* $\Delta(\sigma) < \Delta(\pi)$.

PROOF. We show that schedule σ can be obtained from schedule π by using positive and neutral interchanges only. Compare the two schedules, starting at the end. Suppose the first difference between the schedules occurs in the kth position; J_i occupies the kth position in σ, while job J_j occupies the kth position in π. Because of the choice of J_i and J_j in Algorithm I, we have $p_i \geqslant p_j$; the interchange of J_i and J_i in π is therefore positive or neutral. We proceed in this way until we reach schedule σ. As $\Sigma C_i(\sigma) < \Sigma C_i(\pi)$, at least one of the interchanges must have been positive. Lemma 1 yields the desired result. □

THEOREM 10. *The number of Pareto-optimal schedules is bounded by* $\frac{1}{2}n(n-1)+1$, *and this bound is tight.*

PROOF. The first part follows immediately from Theorem 9. For the second part, consider the following instance of the $1 \mid\mid F(\Sigma C_i, L_{\max})$ problem: there are n jobs with processing times

$p_i = n - 2 + i$ and due dates $d_i = \Sigma_{j=i}^{n} p_j + n - i$, for $i = 1, \ldots, n$. This example generates $\frac{1}{2}n(n-1)+1$ Pareto-optimal schedules.

COROLLARY 1. *The* $1 \,|\,|\, F(\Sigma C_i, \gamma_{\max})$ *problem is solvable in* $\min\{O(n^4), O(n^3(\log n + \log p_{\max}))\}$ *time.*

PROOF. Emmons' algorithm requires $O(n^2)$ time to solve $1 \,|\, \gamma_{\max} \leq \gamma \,|\, \Sigma C_i$. An alternative is to determine the induced deadlines, which requires $O(\log(\Sigma p_i))$ time, and to apply Smith's algorithm subsequently. There are $O(n^2)$ of such problems to be solved. □

COROLLARY 2. *The* $1 \,|\,|\, F(\Sigma C_i, L_{\max})$ *problem is solvable in* $O(n^3)$ *time.*

PROOF. First, note that an upper bound L on the maximum lateness induces a deadline $\bar{d}_i = d_i + L$, which is determined in constant time. Furthermore, in view of Smith's algorithm, it suffices to sort the deadlines only once, since a value-change of L does not affect the order of the deadlines. Once the processing times and deadlines have been sorted, Algorithm II can be implemented to run in linear time. □

4. MINIMIZING TOTAL COMPLETION TIME AND MAXIMUM EARLINESS

In this section, we consider the $1 \,|\,|\, F(\Sigma C_i, E_{\max})$ problem. As $E_{\max}$ is a nonregular performance measure, we additionally assume that all jobs are scheduled in the interval $[0, \Sigma p_i]$, without machine idle time.

It is evident that in each Pareto-optimal schedule σ we have $E^*_{\max} \leq E_{\max}(\sigma) \leq E_{\max}(SPT)$, and $\Sigma C^*_i \leq \Sigma C_i(\sigma) \leq \Sigma C_i(MST)$. The ties in the SPT and MST schedule are settled in order to minimize slack time and processing time, respectively. Observe that an upper bound E on $E_{\max}$ induces for each job J_i a release time $r_i = \max\{0, d_i - p_i - E\}$. The associated value of ΣC_i can then be computed by minimizing total completion time subject to release times and no machine idle time. Lenstra et al. (1977), however, prove that $1 \,|\, r_i, nmit \,|\, \Sigma C_i$, where *nmit* denotes the no-machine-idle-time restriction, is $\mathcal{NP}$-hard in the strong sense (Garey and Johnson, 1979).

Therefore, we make the additional assumption that preemption of jobs is allowed. This is an important relaxation, since the $1 \,|\, pmtn, r_i \,|\, \Sigma C_i$ problem is solvable in $O(n \log n)$ time by Baker's algorithm (1974): *always keep the machine assigned to the available job with minimum remaining processing time.* Note that this algorithm always generates a schedule without machine idle time if $E \geq E^*_{\max}$.

The introduction of preemption has also a less convenient effect. *Any* value of $E_{\max}$ in the range $[E^*_{\max}, E_{\max}(SPT)]$ is now attainable, and therefore corresponds to a Pareto-optimal point. Since $E_{\max}(SPT) - E^*_{\max} \leq \Sigma p_i$, there is a pseudo-polynomial number of Pareto-optimal schedules. These $O(\Sigma p_i)$ Pareto-optimal schedules are generated by the following algorithm.

ALGORITHM III

Step 1. Let $E^{(1)} \leftarrow E_{\max}(SPT)$ and $k \leftarrow 1$.

Step 2. Solve $1 \,|\, pmtn, r_i = d_i - p_i - E^{(k)} \,|\, \Sigma C_i$, giving the kth Pareto-optimal schedule $\sigma^{(k)}$.

Step 3. $k \leftarrow k+1$; $E^{(k)} \leftarrow E^{(k-1)}-1$. If $E^{(k)} \geqslant E^*_{\max}$, go to Step 2, else stop.

COROLLARY 3. *The* $1\,|pmtn,nmit\,|\,F(\Sigma C_i, E_{\max})$ *problem is solvable in* $O(n\,\Sigma p_i \log n)$ *time.*

The next theorem stipulates that the problem is not solvable in polynomial time, unless $\mathcal{P}=\mathcal{NP}$. The proof follows from a reduction from the *Hamiltonian Cycle Problem* due to Schrijver; see Hoogeveen (1990).

THEOREM 11. *The* $1\,|pmtn,nmit\,|\,F(\Sigma C_i, E_{\max})$ *problem is* $\mathcal{NP}$*-hard.*

In the remainder of this section, we investigate the linear variant $1\,|pmtn,nmit\,|\,\alpha_1 \Sigma C_i + \alpha_2 E_{\max}$. As we will see, the linearity of the composite objective function brightens the situation. We have to determine only the extreme points in the range $[E^*_{\max}, E_{\max}(SPT)]$. We define $\sigma(E)$ as the schedule obtained by Baker's algorithm for the $1\,|pmtn,\, E_{\max} \leqslant E\,|\,\Sigma C_i$ problem.

LEMMA 2. *An upper bound* E *on* $E_{\max}$ *can only correspond to an extreme point with respect to the criteria* $E_{\max}$ *and* ΣC_i *if there are two jobs* J_k *and* J_l *such that* $S_l(\sigma(E)) \geqslant C_k(\sigma(E))$, *while* $S_l(\sigma(E-1)) < C_k(\sigma(E-1))$.

PROOF. Let $\sigma(E)$ be the schedule corresponding to the extreme point $(E, \Sigma C_i(\sigma(E)))$. Compare the schedules $\sigma(E-1)$ and $\sigma(E)$. Define a *complete interchange* as an interchange of two jobs J_k and J_l such that $S_l(\sigma(E)) \geqslant C_k(\sigma(E))$ and $S_l(\sigma(E-1)) < C_k(\sigma(E-1))$. Suppose that no complete interchange took place in $\sigma(E)$ in comparison with $\sigma(E-1)$. Furthermore, suppose that $\Sigma C_i(\sigma(E-1)) - \Sigma C_i(\sigma(E)) = \Delta$. As no complete interchange took place, we must have that $\Sigma C_i(\sigma(E)) - \Sigma C_i(\sigma(E+1)) \geqslant \Delta$. From this, it follows that $\sigma(E)$ cannot be an extreme schedule. □

Note that, if the first job J_i in the schedule $\sigma(E)$ is preempted and if E is increased by Δ, where Δ is no more than the length of this portion of job J_i, then $C_i(\sigma(E+\Delta)) \geqslant C_i(\sigma(E))$. Furthermore, the portion of the schedule $\sigma(E)$ between times Δ and $C_i(\sigma(E))$ is identical to the portion of the schedule $\sigma(E+\Delta)$ between times 0 and $C_i(\sigma(E))-\Delta$. This observation is used in Algorithm IV, which computes the smallest value $\overline{E} > E$ that may correspond to an extreme point, where E is a given value of $E_{\max}$ and $\sigma(E)$ is the corresponding schedule.

ALGORITHM IV

Step 1. Let $T \leftarrow 0$ and $a_j \leftarrow \infty$ for $j = 1, \ldots, n$.

Step 2. Let J_i be the job that starts at time T. Consider the following two cases:

(a) J_i is a preempted job. Then a_i is equal to the length of this portion of J_i. Let J_l be the first job that starts after time $C_i(\sigma(E))$ with $p_l \geqslant a_i$. Set $T \leftarrow S_l(\sigma(E))$.

(b) J_i is not a preempted job. Then $a_i \leftarrow \min\{d_j - p_j - E - S_i(\sigma(E)) \mid J_j \in J\}$, where J denotes the set of the jobs for which $d_j - p_j - E > S_i(\sigma(E))$ and $p_i > p_j$. If $J = \varnothing$, then $a_i \leftarrow \infty$. Set $T \leftarrow C_i(\sigma(E))$.

Step 3. If $T < \Sigma p_i$, go to Step 2.

Step 4. Put $\overline{E} \leftarrow \min_i\{a_i\} + E$. Stop.

THEOREM 12. *All values E that may correspond to an extreme point* $(E, \Sigma C_i(\sigma(E))$ *are generated by the iterative application of Algorithm IV.*

PROOF. Let $\overline{E}$ be the $E_{\max}$-value of an extreme point. From Lemma 2, it follows that the increase in maximum earliness from the previous extreme point to $\overline{E}$ must have led to the complete interchange of some job J_i with some job J_j. Suppose Algorithm IV initialized with $E_1 < \overline{E}$ generates $E_2 > \overline{E}$. This implies that the start time of J_j in $\sigma(E)$ was not considered in Step 2. This could take place in Step 2(a) only if J_j was scheduled between the first and the last portion of some preempted job J_k. In this case, as we have observed before, the interchange of J_i and J_j could not have taken place. □

We prove that the number of values E of $E_{\max}$ generated through Algorithm IV is polynomially bounded, thereby establishing that $1\,|pmtn,nmit\,|\,\alpha_1\Sigma C_i + \alpha_2 E_{\max}$ is solved in polynomial time. We define for a given schedule σ

$$\delta_{ij}(\sigma) = \begin{cases} 1 & \text{if } S_i(\sigma) < S_j(\sigma) \text{ and } p_i > p_j, \\ 0 & \text{otherwise} \end{cases},$$

and $\Delta(\sigma) = \Sigma_{i,j}\delta_{ij}(\sigma)$. In addition, an interchange of job J_i with either J_j or a portion of J_j is a *positive* interchange if $p_i < p_j$ and if the interchange is complete. An interchange is *neutral* if either $p_i = p_j$ or if the interchange is not complete.

THEOREM 13. *Let* E_1 *and* E_2 *be two* $E_{\max}$*-values generated through Algorithm IV. Then* $\Sigma C_i(\sigma(E_1)) < \Sigma C_i(\sigma(E_2))$ *implies* $\Delta(\sigma(E_1)) < \Delta(\sigma(E_2))$.

PROOF. We show that schedule $\sigma(E_1)$ can be obtained from the schedule $\sigma(E_2)$ through positive and neutral interchanges only. Compare $\sigma(E_1)$ and $\sigma(E_2)$ with respect to each unit of time, starting at time zero. Suppose the first difference occurs at time T_1. Let job J_i be executed from time T_1 to time T_2 in schedule $\sigma(E_1)$. Adjust schedule $\sigma(E_2)$ by applying interchanges such that $\sigma(E_2)$ is identical to $\sigma(E_1)$ from time 0 to time T_2. Since $\sigma(E_1)$ and $\sigma(E_2)$ were obtained by Baker's rule, the processing time of J_i is smaller than or equal to the remaining processing time of each job J_j that is processed from time T_1 to time T_2 in $\sigma(E_2)$. Therefore, the interchanges needed to adjust schedule $\sigma(E_2)$ are either positive or neutral.

This argument can be repeated until $\sigma(E_1)$ and $\sigma(E_2)$ are identical. As $\Sigma C_i(\sigma(E_1)) < \Sigma C_i(\sigma(E_2))$, at least one of the interchanges must have been positive. Application of Lemma 1 yields the desired result. □

COROLLARY 4. *If preemption is allowed, then the number of extreme schedules with respect to* $E_{\max}$ *and* ΣC_i *is bounded by* $\frac{1}{2}n(n-1)+1$.

PROOF. The maximum number of possible positive interchanges is $\frac{1}{2}n(n-1)$. Theorem 13 then gives the desired result. It is yet an open question whether this bound is tight. □

COROLLARY 5. *The* $1\,|pmtn\,|\,\alpha_1\Sigma C_i+\alpha_2 E_{\max}$ *problem is solvable in* $O(n^4)$ *time.*

THEOREM 14. *If* $\alpha_1=\alpha_2$, *then there exists a nonpreemptive schedule that is optimal for the* $1\,|pmtn\,|\,\alpha_1\Sigma C_i+\alpha_2 E_{\max}$ *problem. If* $\alpha_1>\alpha_2$, *then any optimal schedule for the* $1\,|pmtn\,|\,\alpha_1\Sigma C_i+\alpha_2 E_{\max}$ *problem is nonpreemptive.*

PROOF. Suppose the optimal schedule contains a preempted job. Start at time 0 and find the first preempted job J_i immediately scheduled before some nonpreempted job J_j. Consider the schedule obtained by interchanging job J_j and this portion of job J_i. If the length of the portion of job J_i is Δ, then E_j has been increased by Δ, while C_j has been decreased by Δ. As $\alpha_1=\alpha_2$, the interchange does not increase the objective value. The argument can be repeated until a nonpreemptive schedule remains. In case $\alpha_1>\alpha_2$, then such an interchange would decrease the objective value contradicting the optimality. □

COROLLARY 6. *If* $\alpha_1\geqslant\alpha_2$, *then the* $1\,|\,|\,\alpha_1\Sigma C_i+\alpha_2 E_{\max}$ *problem is solvable in* $O(n^4)$ *time.*

5. MINIMIZING MAXIMUM LATENESS AND MAXIMUM EARLINESS

The analysis of the $1\,|\,|\,F(L_{\max},E_{\max})$ problem is beyond the scope of this paper. Hoogeveen (1990) shows that $1\,|nmit\,|F(L_{\max},E_{\max})$ and $1\,|\,|\,\alpha_1 L_{\max}+\alpha_2 E_{\max}$ are solved in $O(n^2\log n)$ time. Instead, we consider the situation in which the jobs may be preempted, that is, the $1\,|pmtn\,|\,F(L_{\max},E_{\max})$ problem. As $E_{\max}$ is a nonregular performance measure, we impose again the additional restriction that all jobs are processed in the interval $[0,\Sigma p_i]$.

Observe now that each Pareto-optimal schedule must have a maximum lateness that ranges from $L^*_{\max}$ to $\bar{L}_{\max}$, where $\bar{L}_{\max}$ is the outcome of the $1\,|pmtn,\,E_{\max}\leqslant E^*_{\max}\,|\,L_{\max}$ problem. As seen before, the condition $E_{\max}\leqslant E^*_{\max}$ induces a release date for each job J_i. This problem can then be solved by Baker's algorithm (Baker, 1974) for the $1\,|pmtn,\,r_i\,|\,L_{\max}$ problem. The algorithm *always keeps the machine assigned to the available job with the smallest due date.*

In the same fashion, the maximum earliness of each Pareto-optimal point ranges from $E^*_{\max}$ to $\bar{E}_{\max}$, where $\bar{E}_{\max}$ is the outcome of the $1\,|\,nmit,pmtn,\,L_{\max}\leqslant L^*_{\max}\,|\,E_{\max}$ problem. Since this problem reduces to $1\,|\,nmit,pmtn,\,d_i\,|\,E_{\max}$, it is solved by the rule to *keep the machine assigned to the available job with the largest value of* d_i-p_i, *working from time* Σp_i *forward.*

We give a simple algorithm to generate all Pareto-optimal schedules for the $1\,|\,nmit,pmtn\,|\,F(L_{\max},E_{\max})$ problem.

ALGORITHM V

Step 1. Solve the $1\,|\,nmit,pmtn,\,L_{\max}\leqslant L^*_{\max}\,|\,E_{\max}$ problem, which produces the value $\bar{E}_{\max}$.

Step 2. Let $E^{(1)}\leftarrow\bar{E}_{\max}$ and $k\leftarrow 1$.

Step 3. Solve $1\,|pmtn,\,E_{\max}\leqslant E^{(k)}\,|\,L_{\max}$. This produces the kth Pareto-optimal schedule $\sigma^{(k)}$ and associated Pareto-optimal point $(L_{\max}(\sigma^{(k)}),E_{\max}(\sigma^{(k)}))$.

Step 4. $k\leftarrow k+1$, $E^{(k)}\leftarrow E^{(k-1)}-1$. If $E^{(k)}\geqslant E^*_{\max}$, go to Step 3, else stop.

COROLLARY 7. *The* $1 \mid nmit,pmtn \mid F(L_{\max}, E_{\max})$ *problem is solvable in* $O(n\Sigma p_i \log n)$ *time.*

THEOREM 15. *The* $1 \mid nmit,pmtn \mid F(L_{\max}, E_{\max})$ *problem is* $\mathcal{NP}$*-hard.*

PROOF. We assert that the proof proceeds along the same lines as the proof for $1 \mid nmit,pmtn \mid F(\Sigma C_j, E_{\max})$. □

THEOREM 16. *Each Pareto-optimal schedule* $\sigma^{(k)}$ *generated through Algorithm V satisfies* $L_{\max}^{(k)} + E_{\max}^{(k)} = E_{\max}^{*} + \overline{L}_{\max}$.

PROOF. Consider the $1 \mid pmtn, E_{\max} \leqslant E \mid L_{\max}$ and the $1 \mid pmtn, E_{\max} \leqslant E+1 \mid L_{\max}$ problem for some value E, with $E_{\max}^{*} \leqslant E < \overline{E}_{\max}$. Let the optimal schedules be σ and π, respectively. Evidently, we have $L_{\max}(\sigma) = L_{\max}(\pi) - \Delta$ for some $\Delta \geqslant 0$. We show $\Delta = 1$ for any value of E within this range. Consider the schedule σ and let J_i be the first job in σ with $L_i(\sigma) = L_{\max}(\sigma)$. As $L_{\max}(\sigma) > L_{\max}(EDD)$, there must be a job scheduled before J_i with greater due date. If E is increased to $E+1$, then J_i is completed one unit of time earlier. Furthermore, a portion of length one of some job J_k, having $d_k > d_i$ and starting before J_i in σ is transferred to a position after J_i in π. Now consider $C_k(\pi)$. It is obvious that $C_k(\pi) = C_l(\sigma)$, where J_l is the last job in σ with $d_l < d_k$, or, if J_k is already preempted in σ and followed by some job J_h with $d_h \geqslant d_k$, then $C_k(\pi) = C_k(\sigma)$. In both cases we have $L_k(\pi) < L_{\max}(\sigma)$. If there are more jobs in σ with lateness equal to $L_{\max}(\sigma)$, then the above procedure can be repeated. □

Theorem 16 implies that all Pareto-optimal points lie on a straight line. Hence, the $1 \mid nmit,pmtn \mid \alpha_1 E_{\max} + \alpha_2 L_{\max}$ problem has only two extreme points, namely, $(\overline{L}_{\max}, E_{\max}^{*})$ and $(L_{\max}^{*}, \overline{E}_{\max})$.

COROLLARY 8. *The* $1 \mid nmit,pmtn \mid \alpha_1 E_{\max} + \alpha_2 L_{\max}$ *problem is solvable in* $O(n \log n)$ *time.*

ACKNOWLEDGEMENT
The authors would like to thank Jan Karel Lenstra for his helpful comments.

REFERENCES

K.R. BAKER (1974). *Introduction to Sequencing and Scheduling*, Wiley, New York.

H. EMMONS (1975). A note on a scheduling problem with dual criteria. *Naval Research Logistics Quarterly 22*, 615-616.

M.R. GAREY AND D.S. JOHNSON (1979). *Computers and Intractability: a Guide to the Theory of NP-Completeness*, Freeman, San Fransisco.

R.L. GRAHAM, E.L. LAWLER, J.K. LENSTRA AND A.H.G. RINNOOY KAN (1979). Optimization and approximation in deterministic sequencing and scheduling: a survey. *Annals of Discrete Mathematics 5*, 287-326.

S.K. GUPTA AND T. SEN (1984). Minimizing the range of lateness of a single machine. *Journal of the Operational Research Society 35*, 853-857.

J.A. HOOGEVEEN (1990). *On the minimization of maximum earliness and maximum lateness*, Report BS-R9001, Centre for Mathematics and Computer Science, Amsterdam.

J.A. HOOGEVEEN AND S.L. VAN DE VELDE (1990). *A new lower bound approach for single-machine multicriteria scheduling*, Report BS-R90xx, Centre for Mathematics and Computer Science, Amsterdam.

J.R. JACKSON (1955). *Scheduling a production line to minimize maximum tardiness*, Research Report 43, Management Sciences Research Project, UCLA.

E.L. LAWLER (1973). Optimal sequencing of a single machine subject to precedence constraints. *Management Science 19*, 544-546.

J.K. LENSTRA, A.H.G. RINNOOY KAN AND P. BRUCKER (1977). Complexity of machine scheduling problems. *Annals of Discrete Mathematics 1*, 343-362.

R.T. NELSON, R.K. SARIN AND R.L. DANIELS (1986). Scheduling with multiple performance measures: the one-machine case. *Management Science 32*, 464-479.

T. SEN AND S.K. GUPTA (1983). A branch-and-bound procedure to solve a bicriterion scheduling problem. *IIE Transactions 15*, 84-88.

T. SEN, F.M.E. RAISZADEH AND P. DILEEPAN (1988). A branch-and-bound approach to the bicriterion scheduling problem involving total flow time and range of lateness. *Management Science 34*, 254-260.

J.G. SHANTHIKUMAR (1983). Scheduling *n* jobs on one machine to minimize the maximum tardiness with minimum number tardy. *Computers and Operations Research 10*, 255-266.

W.E. SMITH (1956). Various optimizers for single-stage production. *Naval Research Logistics Quarterly 1*, 59-66.

M. TEGZE AND M. VLACH (1988). Improved bounds for the range of lateness on a single machine. *Journal of the Operational Research Society 39*, 675-680.

L.N. VAN WASSENHOVE AND F. GELDERS (1980). Solving a bicriterion scheduling problem. *European Journal of Operational Research 4*, 42-48.