1991

J.A. Hoogeveen

Single-machine scheduling to minimize a function of K maximum cost criteria

Department of Operations Research, Statistics, and System Theory   Report BS-R9113   May

CWI, nationaal instituut voor onderzoek op het gebied van wiskunde en informatica

CWI is the research institute of the Stichting Mathematisch Centrum, which was founded on February 11, 1946, as a non-profit institution aiming at the promotion of mathematics, computer science, and their applications. It is sponsored by the Dutch Government through the Netherlands organization for scientific research (NWO).

Single-Machine Scheduling to Minimize a Function of K Maximum Cost Criteria

J.A. Hoogeveen
CWI
P.O. Box 4079, 1009 AB Amsterdam
The Netherlands

A number of jobs has to be scheduled on a single machine, which can handle no more than one job at a time. Each job requires processing during a given positive uninterrupted time. For each job, there are K arbitrary non-decreasing penalty functions. The quality of a schedule is measured by K performance criteria, the kth one being given by the maximum value of the kth penalty function that is attained by any job. The problem is to find the set of Pareto optimal points with respect to these performance criteria. We present an algorithm for this problem that is polynomial for fixed K. We also show that these algorithms are still applicable if precedence constraints exist between the jobs or if all penalty functions are non-increasing in the job completion times.

1980 Mathematics Subject Classification (1985 Revision): 90B35.
Key Words & Phrases: single-machine scheduling, multicriteria scheduling, Pareto optimal points.

1. Introduction

Since the introduction of scheduling theory in the 1950's, most research has been concentrated on single-criterion optimization. In real-life problems, however, multiple and usually conflicting criteria play a role. There are two methods to cope with conflicting criteria. The first one is *hierarchical* minimization. The performance criteria $f^1, \ldots, f^K$ are indexed in order of decreasing importance. First, f^1 is minimized. Next, f^2 is minimized subject to the constraint that the schedule has minimal f^1 value. Then, f^3 is minimized subject to the constraint that the values for f^1 and f^2 are equal to the values determined in the previous step; and so on. The first results on multicriteria scheduling (e.g., Smith, 1956) concern this approach. The second method is *simultaneous* minimization. The criteria are aggregated into a single composite objective function $P(f^1, \ldots, f^K)$, which is then minimized.

In this paper, we choose the second approach. We address the following single-machine multicriteria scheduling problem. A set of n independent jobs has to be scheduled on a single machine, which can handle no more than one job at a time. The machine is assumed to be continuously available from time 0 onwards. Job J_i $(i = 1, \ldots, n)$ requires processing during a given positive uninterrupted time p_i. A *schedule* σ defines for each job J_i its completion time $C_i(\sigma)$ such that the jobs do not overlap in their execution. The cost of completing J_i $(i = 1, \ldots, n)$ is measured by K penalty functions f_i^k $(k = 1, \ldots, K)$; each of these penalty functions is assumed to be non-decreasing in the job completion time. Given a schedule σ, the penalty functions induce K performance criteria $f_{\max}^k(\sigma)$ $(k = 1, \ldots, K)$, defined as $f_{\max}^k(\sigma) = \max_{1 \leqslant i \leqslant n} f_i^k(C_i(\sigma))$, respectively. Given a function $P: \mathbb{R}^K \rightarrow \mathbb{R}$, we wish to find a

Report BS-R9113
CWI
P.O. Box 4079, 1009 AB Amsterdam, The Netherlands

schedule σ that minimizes $P(f^1_{\max}(\sigma), \ldots, f^K_{\max}(\sigma))$. We additionally assume that P is non-decreasing in each of its arguments. We denote this problem by $1 \,||\, P(f^1_{\max}, \ldots, f^K_{\max})$.

Due to this additional assumption, we know that there is a *Pareto optimal* point with respect to $(f^1_{\max}, \ldots, f^K_{\max})$ in which P attains the optimum. A schedule σ corresponds to a Pareto optimal point if there is no feasible schedule π with $f^k_{\max}(\pi) \leqslant f^k_{\max}(\sigma)$, for $k = 1, \ldots, K$, where at least one of the inequalities is strict; in this case, we say that σ is not dominated.

The organization of the paper is as follows. In Section 2, we recall Lawler's algorithm (Lawler, 1973) for $1 \,|\, prec \,|\, f_{\max}$, where the acronym *prec* indicates that precedence constraints have been specified; that is, for each job J_i $(i = 1, \ldots, n)$ there is a set of jobs that have to precede J_i and a set of jobs that have to succeed J_i in any feasible schedule. Furthermore, we show that we can solve $1 \,|\, \bar{d}_j, prec \,|\, f_{\max}$ by adjusting Lawler's algorithm appropriately, where $\bar{d}_j$ indicates that for each job a deadline on the completion time has been specified. In Section 3, we present an $O(n^4)$ time algorithm to determine all Pareto optimal points for the two-criteria problem. In Section 4, we analyze the three-criteria problem, and show how this analysis can be extended to the K-criteria problem, for any fixed $K \geqslant 4$. Finally, in Section 5, we consider two problems that are solved analogously. The first problem allows precedence constraints; the second one has *non-increasing* penalty functions.

2. LAWLER'S ALGORITHM TO MINIMIZE MAXIMUM COST FOR ONE CRITERION

Lawler (1973) presented an $O(n^2)$ algorithm to solve $1 \,|\, prec \,|\, f_{\max}$. The algorithm is based upon the following observation. Let S denote the subset of jobs that may be processed last, let T denote the sum of the processing times of all jobs, and let J_k be a job in S such that $f_k(T) = \min_{j \in S} \{f_j(T)\}$. Then there exists an optimal schedule in which J_k is last.

LAWLER'S ALGORITHM

(0) $T \leftarrow \Sigma^n_{j=1} p_j$; $\mathfrak{J} \leftarrow \{J_1, \ldots, J_n\}$.

(1) Determine the set $\mathfrak{U}$ containing the jobs that have no successors in $\mathfrak{J}$.

(2) Choose from $\mathfrak{U}$ the job J_j that has minimal $f_j(T)$ value, settling ties arbitrarily; J_j is processed from time $T - p_j$ to time T.

(3) $T \leftarrow T - p_j$; $\mathfrak{J} \leftarrow \mathfrak{J} - \{J_j\}$.

(4) If $\mathfrak{J} \neq \varnothing$, then go to Step 1; otherwise, stop.

THEOREM 1. *Lawler's algorithm solves* $1 \,|\, prec \,|\, f_{\max}$.

PROOF. Let σ be the schedule obtained by Lawler's algorithm, and let $\bar{\sigma}$ be an optimal schedule for $1 \,|\, prec \,|\, f_{\max}$. Compare both schedules, starting at the end. Suppose that the first difference occurs at the kth position; let J_i occupy the kth position in σ. Adjust $\bar{\sigma}$ by assigning J_i to the kth position; the sequence of the other jobs stays the same. The new schedule $\bar{\sigma}$ is feasible; J_i can be assigned to that position as σ is feasible, and the sequence of the other jobs has not been changed. Furthermore, its cost has not been increased; J_i was chosen by Lawler's algorithm, so it must have minimal cost among the unassigned jobs that could be scheduled in that position. Proceed in the same way until σ and $\bar{\sigma}$ are identical, implying that $f_{\max}(\sigma) \leqslant f_{\max}(\bar{\sigma})$. Hence, σ is optimal. □

Lawler's algorithm is easily adjusted to deal with $1|\bar{d}_j, prec|f_{\max}$. If a job J_k is a candidate for the last position, then we have to check whether J_k has no successors and $\bar{d}_j \geqslant T$. Hence, the set $\mathcal{U}$ contains the jobs that have no successors in $\mathcal{J}$ and that have a deadline greater than or equal to T. Alternatively, we can incorporate the deadlines by redefining $f_j(T) \leftarrow \infty$ for $T > \bar{d}_j$ $(j = 1, \ldots, n)$ and apply Lawler's algorithm to the adjusted $1|prec|f_{\max}$ problem.

The deadlines do not have to be given explicitly, but may be induced by given upper bounds on other criteria. For example, if g_i is a non-decreasing penalty function, for $i = 1, \ldots, n$, then the constraint $g_{\max} \leqslant G$ induces a deadline for each job J_i.

3. Analysis of the two-criteria problem

For notational convenience, we denote the penalty functions for J_i $(i = 1, \ldots, n)$ by f_i and g_i, respectively. Correspondingly, the maximum cost criteria are called $f_{\max}$ and $g_{\max}$, respectively. There are basically two ways to deal with the $1||P(f_{\max}, g_{\max})$ problem. The first one is to solve it directly, for instance through branch-and-bound. The second one is to solve it in a roundabout way by determining the Pareto optimal set, that is, the set of points that are Pareto optimal with respect to $(f_{\max}, g_{\max})$, and then selecting the one that minimizes $P(f_{\max}, g_{\max})$. We take the second option. From now on, whenever we refer to the problem $1||P(f^1_{\max}, \ldots, f^K_{\max})$, it is assumed that we are going to determine all Pareto optimal points with respect to $(f^1_{\max}, \ldots, f^K_{\max})$. For instance, $1|\bar{d}_j|P(f_{\max}, g_{\max})$ denotes the problem of determining all Pareto optimal points with respect to $(f_{\max}, g_{\max})$ subject to deadlines.

In order to determine the Pareto optimal set, we apply the following strategy. We start by solving $1||f_{\max}$; this yields the first value F that corresponds to a candidate Pareto optimal point (F, G). Next, we determine the corresponding value G by solving $1|f_{\max} \leqslant F|g_{\max}$ through Lawler's algorithm. Then, we determine the next larger value F that corresponds to a possibly Pareto optimal point (F, G), solve $1|f_{\max} \leqslant F|g_{\max}$ to obtain G, and so on.

There are two difficulties with the application of this strategy. The first one is how is the next value of F determined. The second one concerns the question of how many of these values are to be computed before all Pareto optimal points have been found.

We start by addressing the first problem. Let σ be the schedule obtained by solving $1|f_{\max} \leqslant F|g_{\max}$ through Lawler's algorithm, and let J_j be a job that attains $g_{\max}(\sigma)$, that is, $g_j(C_j(\sigma)) = \max_{1 \leqslant i \leqslant n} g_i(C_i(\sigma))$. As g_j is non-decreasing, a Pareto optimal point with smaller $g_{\max}$ value can be obtained only if the completion time of J_j is decreased. Hence, some job J_i that is before J_j in σ and that has $g_i(C_j(\sigma)) < g_{\max}$ has to be completed no earlier than time $C_j(\sigma)$.

This observation provides the basis for our algorithm to determine the increase of F that is necessary to reach a new candidate Pareto optimal point.

Algorithm Next Upper Bound (NUB)

(0) Given a schedule σ obtained by Lawler's algorithm, determine the set $\mathcal{J}$ of jobs that attain $g_{\max}(\sigma)$.

(1) Determine for each $J_j \in \mathcal{J}$ the set $\mathcal{U}_j$ of jobs J_i that are scheduled before J_j in σ and that have $g_i(C_j(\sigma)) < g_{\max}(\sigma)$. If $\mathcal{U}_j = \varnothing$ for some $J_j \in \mathcal{J}$, then $g_{\max}(\sigma)$ cannot be decreased; stop. For each job $J_j \in \mathcal{J}$, define $F_j = \min\{f_i(C_j(\sigma)) | J_i \in \mathcal{U}_j\}$.

(2) The new upper bound F on $f_{\max}$ is the maximum of the values F_j.

THEOREM 2. *Let $(\hat{F},\hat{G})$ be a Pareto optimal point with respect to $(f_{\max}, g_{\max})$, and let σ be the corresponding schedule. Let F be the new upper bound on $f_{\max}$ that is obtained by applying Algorithm* NUB, *given σ. There is no Pareto optimal point corresponding to a value $\tilde{F}$, with $F > \tilde{F} > \hat{F}$.* □

A decrease of C_j does not necessarily induce a decrease in $g_j(C_j)$, and hence the new upper bound F does not necessarily correspond to a Pareto optimal point. The remaining question is how many values F are determined by Algorithm NUB, before all Pareto optimal points have been found.

THEOREM 3. *The Algorithm* NUB *determines at most $n(n-1)/2$ values F.*

PROOF. Every new value F obtained by applying Algorithm NUB to σ corresponds to a combination of two jobs $\{J_i, J_j\}$, where $g_j(C_j(\sigma)) = g_{\max}(\sigma)$ and $f_i(C_j(\sigma)) = F$. We will show that every upper bound value F that is obtained by Algorithm NUB corresponds to a different combination of jobs.

Define f^* and g^* as the outcome of $1||f_{\max}$ and $1||g_{\max}$, respectively. Let σ_1 be the schedule obtained by Lawler's algorithm when solving $1|f_{\max} \leqslant f^*|g_{\max}$. For $a = 1, \ldots, A$, apply Algorithm NUB to schedule σ_a to obtain F^{a+1}, and determine σ_{a+1} by solving $1|f_{\max} \leqslant F^{a+1}|g_{\max}$; A is such that $g_{\max}(\sigma_A) = g^*$. Suppose that the combination $\{J_i, J_j\}$ corresponds to both F^{a+1} and F^{b+1}, with $a < b$. Without loss of generality, let J_j attain $g_{\max}(\sigma_a)$; $f_i(C_j(\sigma_a)) = F^{a+1}$. We have to consider two cases: either J_i or J_j attains $g_{\max}(\sigma_b)$. First, suppose that $g_j = g_{\max}(\sigma_b)$; $f_i(C_j(\sigma_b)) = F^{b+1}$. As $g_j(C_j(\sigma_a)) = g_{\max}(\sigma_a) \geqslant g_{\max}(\sigma_b) = g_j(C_j(\sigma_b))$, we must have $C_j(\sigma_a) \geqslant C_j(\sigma_b)$. This implies that J_i is allowed to be completed at time $C_j(\sigma_b)$ when σ_b is constructed, because $F^b > F^{a+1}$. As Lawler's algorithm selected J_j to be completed at time $C_j(\sigma_b)$, we must have that either J_i had already been scheduled or $g_j(C_j(\sigma_b)) \leqslant g_i(C_j(\sigma_b))$. In both cases, Algorithm NUB will not take J_i into consideration, when applied to σ_b. In the same fashion, we prove that J_j will not be taken into consideration by the algorithm if J_i attains $g_{\max}(\sigma_b)$. Hence, every pair of jobs (J_i, J_j) corresponds to at most one of the values F obtained by Algorithm NUB. This proves the theorem. □

COROLLARY 1. *The number of Pareto optimal points with respect to $(f_{\max}, g_{\max})$ is at most equal to $n(n-1)/2 + 1$.* □

The following example shows that this bound is tight, even if both maximum cost functions are of the maximum lateness type, that is, $f_i : C_i \to C_i - d_i$, and $g_i : C_i \to C_i - e_i$, for $i = 1, \ldots, n$.

$$d_i = (n-i)(n-i+3)/2, \quad \text{for } i = 1, \ldots, n,$$

$$e_i = i - 1 + \sum_{j=2}^{i} d_{j+1}, \qquad \text{for } i = 1, \ldots, n-1,$$

$$e_n = e_{n-1} + 1,$$

$$p_i = n - i, \qquad \text{for } i = 1, \ldots, n-1.$$

$$p_n = d_n - (n-1)(n-2)/2;$$

It is easy to check that the Pareto optimal schedules for this example are: $(J_n, \ldots, J_2, J_1)$, $(J_n, \ldots, J_1, J_2), \ldots, (J_1, J_n, \ldots, J_3, J_2), \ldots, (J_1, J_2, J_n, \ldots, J_4, J_3), \ldots, (J_1, \ldots, J_n)$.

For sake of completeness, we list the algorithm to determine all Pareto optimal points and the optimal solution value. Its correctness follows from Theorems 1 and 2.

ALGORITHM A

(0) Determine f^* and g^* by solving $1 \mid \mid f_{\max}$ and $1 \mid \mid g_{\max}$, respectively; put $F \leftarrow f^*$.

(1) Solve $1 \mid f_{\max} \leqslant F \mid g_{\max}$; let G denote the outcome. Add (F, G) to the set of Pareto optimal points, unless it is dominated by the previously obtained Pareto optimal point. If $G = g^*$, then go to Step 3.

(2) Determine the next value of F by applying Algorithm NUB to the schedule obtained in the previous step. Go to Step 1.

(3) The Pareto optimal set has been obtained. The $1 \mid \mid P(f_{\max}, g_{\max})$ problem is solved by computing the value of the objective function for each point of the Pareto optimal set, and by choosing the optimum.

The running time of Algorithm A is $O(n^4)$; this is the time needed for solving $O(n^2)$ instances of the $1 \mid f_{\max} \leqslant F \mid g_{\max}$ problem.

4. ANALYSIS OF THE K-CRITERIA PROBLEM

We prove that the K-criteria problem can be solved by solving a polynomial number of $(K-1)$-criteria problems. First, we analyze the three-criteria problem; later on, we show how this analysis can be extended to the K-criteria problem. For notational convenience, the criteria are called $f_{\max}$, $g_{\max}$, and $h_{\max}$, respectively; correspondingly, the penalty functions for J_i $(i = 1, \ldots, n)$ are called f_i, g_i, and h_i, respectively.

Note that each Pareto optimal point (F, G) for $(f_{\max}, g_{\max})$ yields a Pareto optimal (F, G, H) for $(f_{\max}, g_{\max}, h_{\max})$, where H is the outcome of $1 \mid f_{\max} \leqslant F, g_{\max} \leqslant G \mid h_{\max}$, and that each non-Pareto optimal point (F, G) can only correspond to a Pareto optimal point (F, G, H) if H is attractive enough. Note further that, if (F, G, H) is Pareto optimal, then (F, G) is a solution of $1 \mid h_{\max} \leqslant H \mid P(f_{\max}, g_{\max})$. These observations provide the basis for our strategy to solve the three-criteria problem, denoted by $1 \mid \mid P(f_{\max}, g_{\max}, h_{\max})$.

We will determine all Pareto optimal points (F, G, H) for $(f_{\max}, g_{\max}, h_{\max})$ by considering all $h_{\max}$ values that correspond to a candidate Pareto optimal point. The $h_{\max}$ values under consideration lie in the interval $[h^*, \overline{H}]$; h^* is the solution of $1 \mid \mid h_{\max}$, and $\overline{H}$ is an upper bound on the $h_{\max}$ value of any Pareto optimal point that we will establish now. Obviously, $\overline{H}$ should be such that solving $1 \mid h_{\max} \leqslant \overline{H} \mid P(f_{\max}, g_{\max})$ yields the set of Pareto optimal points for $(f_{\max}, g_{\max})$. Hence, $\overline{H}$ is determined by solving $1 \mid f_{\max} \leqslant F, g_{\max} \leqslant G \mid h_{\max}$ for every Pareto optimal point (F, G) for $(f_{\max}, g_{\max})$ and selecting the maximum. If we want to determine new Pareto optimal points (F, G, H), then we have to decrease the upper bound on $h_{\max}$, such that at least one of the currently determined Pareto optimal points is eliminated. This leads to the following outline for an algorithm to determine all Pareto optimal points for $(f_{\max}, g_{\max}, h_{\max})$.

(0) Determine the set of Pareto optimal points (F,G) with respect to $(f_{\max}, g_{\max})$. For each of these points, compute the corresponding $h_{\max}$ value, say, H. Store these Pareto optimal points (F,G,H) in a set U_1.
(1) Determine $\overline{H}$ as the maximum of the $h_{\max}$ values in U_1. Remove the Pareto optimal points with $h_{\max}$ value equal to $\overline{H}$ and store them in a set U_2.
(2) If $\overline{H} = h^*$, then stop; the set of Pareto optimal points is equal to $U_1 \cup U_2$.
(3) Solve $1 \mid h_{\max} < \overline{H} \mid P(f_{\max}, g_{\max})$, and compute for these points (F,G) the corresponding $h_{\max}$ values H. Add these points (F,G,H) to U_1. Go to 1.

We solve $1 \mid h_{\max} < \overline{H} \mid P(f_{\max}, g_{\max})$ by adjusting Algorithm A such that every solution that is generated by Algorithm A satisfies $h_{\max} < \overline{H}$. As observed before, this is easily achieved by adjusting the penalty functions appropriately.

Before proving that this strategy determines all points (F,G,H) that are Pareto optimal with respect to $(f_{\max}, g_{\max}, h_{\max})$, we prove two preliminary results.

THEOREM 4. *Let (F,G) be an arbitrary Pareto optimal point that is obtained when solving $1 \mid h_{\max} < \overline{H} \mid P(f_{\max}, g_{\max})$; let H be the outcome of $1 \mid f_{\max} \leq F, g_{\max} \leq G \mid h_{\max}$. The point (F,G,H) is Pareto optimal with respect to $(f_{\max}, g_{\max}, h_{\max})$.*

PROOF. Suppose that there exists a Pareto optimal point $(\tilde{F},\tilde{G},\tilde{H})$ that dominates (F,G,H). This implies that $(\tilde{F},\tilde{G})$ is obtained when solving $1 \mid h_{\max} \leq \tilde{H} \mid P(f_{\max}, g_{\max})$. As $\tilde{H} \leq H < \overline{H}$, the constraint $h_{\max} \leq \tilde{H}$ is at least as restrictive as $h_{\max} < \overline{H}$, implying that the point $(\tilde{F},\tilde{G})$ is also obtained when solving $1 \mid h_{\max} < \overline{H} \mid P(f_{\max}, g_{\max})$. Hence, $\tilde{F} = F$ and $\tilde{G} = G$, implying that $\tilde{H} = H$, as both values are equal to the outcome of $1 \mid f_{\max} \leq F, g_{\max} \leq G \mid h_{\max}$. □

COROLLARY 2. *Let (F,G,H) be an arbitrary point with $H < \overline{H}$ that is not found when solving $1 \mid h_{\max} < \overline{H} \mid P(f_{\max}, g_{\max})$. Then there exists a Pareto optimal point $(\tilde{F},\tilde{G},\tilde{H})$ with $\tilde{H} < \overline{H}$ such that $\tilde{F} \leq F$ and $\tilde{G} \leq G$, where at least one of the inequalities is strict.* □

THEOREM 5. *Every Pareto optimal point with respect to $(f_{\max}, g_{\max}, h_{\max})$ is found.*

PROOF. Let (F,G,H) be an arbitrary Pareto optimal point with respect to $(f_{\max}, g_{\max}, h_{\max})$. If (F,G) is Pareto optimal with respect to $(f_{\max}, g_{\max})$, then (F,G,H) is determined at the initialization. Otherwise, there must exist a Pareto optimal point that dominates (F,G,H) with respect to $(f_{\max}, g_{\max})$. Suppose that $(\tilde{F},\tilde{G},\tilde{H})$ is the Pareto optimal point with the smallest $h_{\max}$ value that dominates (F,G,H) with respect to $(f_{\max}, g_{\max})$. Hence, (F,G,H) will be generated as soon as the upper bound on $h_{\max}$ has become smaller than $\tilde{H}$. □

A straightforward implementation of the strategy leads to an $O(n^4 \mid Z \mid)$ time algorithm, where $\mid Z \mid$ denotes the cardinality of the set of Pareto optimal points. The factor $O(n^4)$ stems from solving an $1|f_{\max} \leq F, g_{\max} \leq G \,|\, h_{\max}$ problem for every point (F,G) that is obtained when solving $1|h_{\max} < \overline{H} \,|\, P(f_{\max}, g_{\max})$. Note that we have not yet taken precautions to avoid a point (F,G) being generated more than once. Hence, we may improve the time complexity by determining a quick way to generate every Pareto optimal point only once. This is achieved by generating only the Pareto optimal points that are not present in the current set U_1, when solving $1|h_{\max} < \overline{H} \,|\, P(f_{\max}, g_{\max})$; these are exactly the points that are dominated with respect to $(f_{\max}, g_{\max})$ by a Pareto optimal point (F,G,H) with $H = \overline{H}$, but not by any other Pareto optimal point $(\hat{F},\hat{G},\hat{H})$ in U_1 with $\hat{H} < \overline{H}$. In order to determine only these Pareto optimal points, we derive lower and upper bounds on the $f_{\max}$ value such that a new Pareto optimal point must have a $f_{\max}$ value that is in between. We then search within this region for $f_{\max}$ values that correspond to a possibly Pareto optimal point by applying Algorithm NUB. The schedule we need to start with is obtained simultaneously with the bounds.

Order the Pareto optimal points in the set U_1 lexicographically, that is, put the points in non-decreasing order of $f_{\max}$ value, settling ties according to non-decreasing $g_{\max}$ values. Let (F^1, G^1, H^1) be the last point before (F,G,H) in the list with $H^1 < H$, and let σ_1 be the corresponding schedule. If there is no such point, then F^1 is equal to the outcome of $1|h_{\max} < \overline{H} \,|\, f_{\max}$, G^1 to the outcome of $1|f_{\max} \leq F^1, h_{\max} < \overline{H} \,|\, g_{\max}$, and H^1 to the outcome of $1|f_{\max} \leq F^1, g_{\max} \leq G^1 \,|\, h_{\max}$, respectively; σ^1 is then the corresponding schedule. Let (F^2, G^2, H^2) be the first point after (F,G,H) with $h_{\max}$ value smaller than H. If such a point does not exist, then $F^2 = \infty$.

The new Pareto optimal points are determined by an iterative procedure. Apply Algorithm NUB to σ_1; this yields an $f_{\max}$ value $\hat{F}$. Determine $\hat{G}$ by solving $1|f_{\max} \leq \hat{F}, h_{\max} < H \,|\, g_{\max}$, and $\hat{H}$ by solving $1|f_{\max} \leq \hat{F}, g_{\max} \leq \hat{G} \,|\, h_{\max}$; call the corresponding schedule $\hat{\sigma}$. If $\hat{F} \geq F^2$, then stop; otherwise repeat the above procedure, in which Algorithm NUB is applied to $\hat{\sigma}$.

THEOREM 6. *Let $(\hat{F},\hat{G},\hat{H})$ be an arbitrary Pareto optimal point that is dominated with respect to $(f_{\max}, g_{\max})$ by (F,G,H), but that is not dominated with respect to $(f_{\max}, g_{\max})$ by a point in U_1 with $h_{\max}$ value smaller than H. Then $F \leq \hat{F} < F^2$; these Pareto optimal points $(\hat{F},\hat{G},\hat{H})$ are all determined by the procedure described above.*

PROOF. First, we prove the validity of the bounds on $\hat{F}$. The lower bound F follows by definition; the upper bound follows from the observation that $G^2 < G$ as (F^2, G^2, H^2) is not dominated by (F,G,H) with respect to $(f_{\max}, g_{\max})$, and hence $G^2 < \hat{G}$. As $(\hat{F},\hat{G},\hat{H})$ is not dominated by (F^2, G^2, H^2) with respect to $(f_{\max}, g_{\max})$, $\hat{F}$ must be smaller than F^2.

Second, the point (F^1, G^1) is a solution of $1|h_{\max} < H \,|\, P(f_{\max}, g_{\max})$. Applying Algorithm NUB as described above, starting with σ^1, yields a set of values $\hat{F}$ each corresponding to a possibly Pareto optimal point. None of these points is dominated with respect to $(f_{\max}, g_{\max})$ by a point already in U_1, as $\hat{F}$ is chosen such that $\hat{G} \leq G^1$, while (F^1, G^1, H^1) is not dominated by a point in U_1 with respect to $(f_{\max}, g_{\max})$. □

Note that we have to check whether the $f_{\max}$ value determined by Algorithm NUB corresponds to a Pareto optimal point (F,G) with respect to $(f_{\max}, g_{\max})$ subject to the constraint $h_{\max} < H$, as an increase of the $f_{\max}$ value does not necessarily induce a decrease of the $g_{\max}$ value.

The theorems above show that our strategy can be implemented in such a way that all Pareto optimal points with respect to $(f_{max}, g_{max}, h_{max})$ are found in $O(n^2 |P|)$ time. The $O(n^2)$ component per Pareto optimal point is needed to solve $1|h_{max} < \overline{H}|P(f_{max}, g_{max})$, to solve $1|f_{max} \leqslant F, g_{max} \leqslant G|h_{max}$, to order the Pareto optimal points in U_1 lexicographically, and to determine the maximum of the h_{max} values in U_1. It remains to be shown that $|P|$ is polynomially bounded in n.

THEOREM 7. *There are at most* $n^2(n-1)^2/4 + n(n-1) + 1$ *Pareto optimal points with respect to* $(f_{max}, g_{max}, h_{max})$.

PROOF. Immediately after the initialization, U_1 contains at most $n(n-1)/2 + 1$ points. Every other Pareto optimal point (F, G, H) is dominated with respect to (f_{max}, g_{max}), and hence is generated in the remainder of the algorithm.

Consider an arbitrary point (F, G, H) that is generated from $(\overline{F}, \overline{G}, \overline{H})$. Let $\overline{\sigma}$ be the schedule corresponding to $(\overline{F}, \overline{G}, \overline{H})$, and let σ be the schedule corresponding to (F, G, H). Let h_{max} be attained by J_j in $\overline{\sigma}$. As $\overline{H} > H$, J_j must be completed earlier in σ, and hence, there must be a job J_i preceding J_j in $\overline{\sigma}$ that in σ is completed at or after time $C_j(\overline{\sigma})$. As $F \geqslant \overline{F}$ and $G \geqslant \overline{G}$, we can prove along the same lines as in the proof of Theorem 3 that this combination $\{J_i, J_j\}$ will not occur again, when h_{max} is decreased. Hence, every point obtained in Step 0 dominates at most $n(n-1)/2$ Pareto optimal points with respect to (f_{max}, g_{max}), which implies that there are at most $n^2(n-1)^2/4 + n(n-1) + 1$ Pareto optimal points. □

For sake of completeness, we list the $O(n^6)$ time algorithm to determine all Pareto optimal points. Its correctness follows from Theorems 5 through 9.

ALGORITHM B

(0) Solve $1||P(f_{max}, g_{max})$. Determine for each point (F, G) the corresponding h_{max} value by solving $1|f_{max} \leqslant F, g_{max} \leqslant G|h_{max}$. Store these points in set U_1; determine $\overline{H}$ as the maximum of the h_{max} values of the points in U_1.

(1) Order the points in U_1 lexicographically and let $\overline{H}$ be the maximum h_{max} value. Let (F, G, H) be the first point in the list with h_{max} value equal to $\overline{H}$. Determine the bound F^2 and the schedule σ^1, and solve $1|F \leqslant f_{max} < F^2, h_{max} < \overline{H}|P(f_{max}, g_{max})$ as described on the previous page, and scan the solution set for Pareto optimal points $(\hat{F}, \hat{G})$ with respect to (f_{max}, g_{max}). Determine the corresponding h_{max} value by solving $1|f_{max} \leqslant \hat{F}, g_{max} \leqslant \hat{G}|h_{max}$. Remove (F, G, H) from U_1 and store it in the set U_2. Add the newly obtained points to U_1.

(2) If $\overline{H}$ is greater than the outcome of $1||h_{max}$, then go to Step 1. Otherwise, the union of the sets U_1 and U_2 contains all Pareto optimal points (F, G, H) with respect to $(f_{max}, g_{max}, h_{max})$.

It is easily checked that the strategy that was followed to solve the three-criteria problem can also be applied to solve the K-criteria problem, as the Theorems 4 and 5 and Corollary 2 still hold for the K-criteria case. Unfortunately, Theorem 6 no longer holds, so we can no longer guarantee that each Pareto optimal point is generated only once. We now solve $O(|P|)$ times the problem of determining all Pareto optimal points for a $(K-1)$-criteria problem with a given upper bound on $f_{\max}^K$; for each of the obtained points, we determine the corresponding $f_{\max}^K$ value.

The proof of Theorem 7 can be extended to the K-criteria case, showing that there are at most $(n(n-1)/2+1)^{K-1}$ Pareto optimal points. Therefore, our strategy can be implemented to run in $O(n^{K(K+1)-6})$ time, for $K \geqslant 4$.

5. Related problems

We finally consider the problems $1|prec|P(f_{\max}^1,\ldots,f_{\max}^K)$ and $1||P(g_{\max}^1,\ldots,g_{\max}^K)$, where the functions $g_{\max}^k$ are induced by penalty functions g_j^k that are non-increasing in the job completion times, for $k=1,\ldots,K$. We show that we can solve both problems by Algorithm B by modifying the penalty functions appropriately.

First, we deal with $1|prec|P(f_{\max}^1,\ldots,f_{\max}^K)$. Let $\mathcal{J}_i$ denote the set of jobs J_j that have to succeed J_i in any feasible schedule σ. As $C_i(\sigma) < C_j(\sigma)$ for each job $J_j \in \mathcal{J}_i$, the cost of σ does not increase if at time T ($T \in [0, \Sigma p_j]$) the value of the penalty functions f_i^k ($k=1,\ldots,K$) is set equal to $\max\{f_i^k(T), f_j^k(T)\}$, for each $J_j \in \mathcal{J}_i$. The next theorem shows that the precedence constraints can be handled by adjusting the penalty functions as described above.

THEOREM 8. *The* $1|prec|P(f_{\max}^1,\ldots,f_{\max}^K)$ *problem is solved by adjusting the penalty functions as described above, provided that ties in Lawler's algorithm are settled according to the precedence constraints.*

PROOF. Let g_i^k ($k=1,\ldots,K$) denote the adjusted penalty functions. The proof consists of two parts. First, we have to show that every Pareto optimal point for $1||P(g_{\max}^1,\ldots,g_{\max}^K)$ corresponds to a schedule that is feasible with respect to the precedence constraints. This is guaranteed by the requirement to settle ties in Lawler's algorithm according to the precedence constraints.

Second, we have to prove that the sets of Pareto optimal points for $1|prec|P(f_{\max}^1,\ldots,f_{\max}^K)$ and $1||P(g_{\max}^1,\ldots,g_{\max}^K)$ are the same. Note that a point $(F^1,\ldots,F^K)$ is Pareto optimal with respect to $(f_{\max}^1,\ldots,f_{\max}^K)$ subject to precedence constraints if and only if, for $k=1,\ldots,K$, the outcome of the problem of minimizing $f_{\max}^k$ subject to the constraints $f_{\max}^i \leqslant F^i$ ($i=1,\ldots,K; i \neq k$) and precedence constraints is equal to F^k. Furthermore, a point $(F^1,\ldots,F^K)$ is Pareto optimal with respect to $(g_{\max}^1,\ldots,g_{\max}^K)$ if and only if, for $k=1,\ldots,K$, the outcome of the problem of minimizing $g_{\max}^k$ subject to the constraints $g_{\max}^i \leqslant F^i$ ($i=1,\ldots,K; i \neq k$) is equal to F^k. Hence, if we prove that the problems $1|f_{\max}^1 \leqslant F^1,\ldots,f_{\max}^{K-1} \leqslant F^{K-1}, prec|f_{\max}^K$ and $1|g_{\max}^1 \leqslant F^1,\ldots,g_{\max}^{K-1} \leqslant F^{K-1}|g_{\max}^K$ yield the same outcome, then we are done. To that end, we have to justify the following three claims.

(1) The outcome of the problem $1|f_{\max}^1 \leqslant F^1,\ldots,f_{\max}^{K-1} \leqslant F^{K-1}, prec|f_{\max}^K$ stays the same if we replace the constraints $f_{\max}^k \leqslant F^k$ by $g_{\max}^k \leqslant F^k$, for $k=1,\ldots,K-1$.

(2) The outcome of the problem $1|g_{\max}^1 \leqslant F^1,\ldots,g_{\max}^{K-1} \leqslant F^{K-1}, prec|f_{\max}^K$ stays the same if we replace the objective function $f_{\max}^K$ by $g_{\max}^K$.

(3) The precedence constraints in the problem $1|g_{\max}^1 \leqslant F^1,\ldots,g_{\max}^{K-1} \leqslant F^{K-1}, prec|g_{\max}^K$ are redundant.

Proof of (1). The first claim is proven by showing that the sets of feasible schedules are identical for both problems. The nontrivial part consists of showing that every schedule $\sigma \in \{\sigma \mid f_{\max}^1 \leqslant F^1, \ldots, f_{\max}^{K-1} \leqslant F^{K-1}, prec\}$ has $g_{\max}^k \leqslant F^k$ $(k = 1, \ldots, K-1)$. Suppose to the contrary that there exists a feasible schedule σ with $g_i^k(C_i(\sigma)) > F^k$ for some job J_i, for some k. Then J_i must have a successor J_j such that $g_i^k(C_i(\sigma)) = f_j^k(C_i(\sigma)) \leqslant f_j^k(C_j(\sigma)) \leqslant F^k$, contradicting the assumption.

Proof of (2). The second claim is proven by showing that $f_{\max}^K(\sigma) = g_{\max}^K(\sigma)$ for every feasible schedule σ. By definition, $f_{\max}^K(\sigma) \leqslant g_{\max}^K(\sigma)$. Let $g_{\max}^K$ be attained by J_i; suppose that $f_{\max}^K(\sigma) < g_{\max}^K(\sigma)$. Hence, J_i must have a successor J_j with $f_j^K(C_i(\sigma)) = g_i^K(C_i(\sigma))$. In that case, however, $f_{\max}^K \geqslant f_j^K(C_j(\sigma)) \geqslant f_j^K(C_i(\sigma)) = g_i^K(C_i(\sigma)) = g_{\max}^K$, contradicting the assumption.

Proof of (3). Consider an arbitrary job J_i; let J_j be a successor of J_i. As $g_i^k(T) \geqslant g_j^k(T)$ $(k = 1, \ldots, K-1;\ T = 1, \ldots, \Sigma p_j)$, job J_j will be available for processing if job J_i is. Hence, Lawler's algorithm yields an optimal schedule for $1 \mid g_{\max}^1 \leqslant F^1, \ldots, g_{\max}^{K-1} \leqslant F^{K-1} \mid g_{\max}^K$ that satisfies the precedence constraints, provided that ties are settled according to the precedence constraints. □

Corollary 3. *The* $1 \mid \bar{d}_j, prec \mid P(f_{\max}^1, \ldots, f_{\max}^K)$ *problem can be solved in* $O(n^{2K})$ *time for* $K = 2, 3$, *and in* $O(n^{K(K+1)-6})$ *time for* $K \geqslant 4$. □

The second problem we consider in this section is $1 \mid nmit \mid P(g_{\max}^1, \ldots, g_{\max}^K)$, where the maximum cost functions $g_{\max}^k$ $(k = 1, \ldots, K)$ are induced by penalty functions g_j^k $j = 1, \ldots, n; k = 1, \ldots, K)$ that are non-increasing functions of the job completion times. In order to avoid unbounded solutions, we make the additional assumption that no machine idle time is allowed. This assumption is denoted by the acronym *nmit*, and implies that all jobs are processed in the time interval $[0, \Sigma p_j]$. We show that this problem can be transformed into a problem that fits in the existing framework, and hence, that it is solved in $O(n^{2K})$ time for $K = 2, 3$, and in $O(n^{K(K+1)-6})$ time for $K \geqslant 4$.

Theorem 9. *Lawler's algorithm solves* $1 \mid g_{\max}^1 \leqslant G^1, \ldots, g_{\max}^{K-1} \leqslant G^{K-1}, nmit \mid g_{\max}^K$ *to optimality.*

Proof. Consider an arbitrary instance of the $1 \mid g_{\max}^1 \leqslant G^1, \ldots, g_{\max}^{K-1} \leqslant G^{K-1}, nmit \mid g_{\max}^K$ problem. Now construct the following instance of $1 \mid f_{\max}^1 \leqslant F^1, \ldots, f_{\max}^{K-1} \leqslant F^{K-1} \mid f_{\max}^K$. The processing times are identical for both problems, $f_i^k(T) = g_i^k(\Sigma p_j + p_i - T)$ $(i = 1, \ldots, n;\ k = 1, \ldots, K;\ T = 1, \ldots, \Sigma p_j)$, and $F^k = G^k$, for $k = 1, \ldots, K-1$. Suppose that Lawler's algorithm yields schedule σ for $1 \mid f_{\max}^1 \leqslant F^1, \ldots, f_{\max}^{K-1} \mid f_{\max}^K$. An optimal schedule $\bar{\sigma}$ for $1 \mid g_{\max}^1 \leqslant G^1, \ldots, g_{\max}^{K-1} \leqslant G^{K-1}, nmit \mid g_{\max}^K$ is obtained by reversing σ; $C_i(\bar{\sigma}) = \Sigma p_j + p_i - C_i(\sigma)$ $(i = 1, \ldots, n)$, and hence, $g_i^k(C_i(\bar{\sigma})) = f_i^k(C_i(\sigma))$ $(i = 1, \ldots, n;\ k = 1, \ldots, K)$. This implies that σ is optimal and feasible if and only if $\bar{\sigma}$ is optimal and feasible. □

Corollary 4. *A point* $(F^1, \ldots, F^K)$ *is Pareto optimal with respect to* $(f_{\max}^1, \ldots, f_{\max}^K)$ *if and only if this point is Pareto optimal with respect to* $(g_{\max}^1, \ldots, g_{\max}^K)$. □

From Corollary 4 it follows immediately that we can solve $1 | nmit | P(g_{\max}^1, \ldots, g_{\max}^K)$ by transforming it to an $1 | | P(f_{\max}^1, \ldots, f_{\max}^K)$ problem as described in Theorem 9, and by applying Algorithm B to this instance. As a deadline $\bar{d}_j$ for the $1 | | P(f_{\max}^1, \ldots, f_{\max}^K)$ problem corresponds to a release date r_j, that is, a lower bound on the start time for J_j, $1 | r_j, nmit, prec | P(g_{\max}^1, \ldots, g_{\max}^K)$ is solvable in $O(n^{2K})$ time for $K = 2,3$, and in $O(n^{K(K+1)-6})$ time for $K \geq 4$.

ACKNOWLEDGEMENT

The author is grateful to Jan Karel Lenstra and Steef van de Velde for their helpful comments.

REFERENCES

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

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