McMASTER
UNIVERSITY

MICHAEL G. DeGROOTE
SCHOOL OF BUSINESS

RESEARCH AND
WORKING PAPER
SERIES

SUBSET-RESTRICTED INTERCHANGE FOR DYNAMIC MIN-MAX SCHEDULING PROBLEMS

By

George Steiner and **Paul Stephenson**

Management Science and Information Systems Area
Michael G. DeGroote School of Business
McMaster University,
Hamilton, Ontario, Canada

Working Paper # 418

October, 1996

TER UNIVERSITY
n Street West
, Ontario, Canada L8S 4M4
e (905) 525-9140

SUBSET-RESTRICTED INTERCHANGE FOR DYNAMIC MIN-MAX SCHEDULING PROBLEMS

By

George Steiner and **Paul Stephenson**

Management Science and Information Systems Area
Michael G. DeGroote School of Business
McMaster University,
Hamilton, Ontario, Canada

Working Paper # 418

October, 1996

Subset-Restricted Interchange for Dynamic Min-Max Scheduling Problems*

George Steiner†and Paul Stephenson
Management Science and Information Systems Area
Michael G. DeGroote School of Business
McMaster University, Hamilton, Ontario, Canada

Abstract

The traditional method of pairwise job interchange compares the cost of sequences that differ only in the interchange of two jobs. It assumes that either there are no intermediate jobs (*adjacent pairwise interchange*) or that the interchange can be performed no matter what the intermediate jobs are (*nonadjacent pairwise interchange*). We introduce a generalization that permits the pairwise interchange of jobs *provided* that the intermediate jobs belong to a restricted subset of jobs (*subset-restricted pairwise interchange*).

In general, even if an adjacent interchange relation is a partial order it need not be a precedence order. We introduce a unified theory of dominance relations based on subset-restricted interchange. This yields a precedence order for the class of unconstrained, regular, single machine scheduling problems $1\,/r\,/f_{\max}$. Thus it applies to $1\,/r\,/L_{\max}$, $1\,\big/r,\bar{d}\big/\,C_{\max}$, $1\,/r\,/WL_{\max}$, $1\,/r\,/WC_{\max}$ and other problems. We also show that these problems remain strongly NP-hard for interval-ordered tasks.

1 Introduction

Interchange arguments, that compare the cost of sequences that differ only in the interchange of two jobs, are quite common in the scheduling literature. Some classic results that established the technique are the *rules* of *Johnson* [9], *Smith* [18], and

*This research was supported in part by the Natural Sciences and Engineering Research Council of Canada, under Grant No. OGP0001798.

†Corresponding author. e-mail: steiner@mcmaster.ca

Jackson [8]. If jobs i and j can be interchanged so that i is before j in an optimal sequence, then we will say that i is *preferred* to j. If these preference relations are transitive, then we have a *preference order* on the jobs. The above mentioned rules are examples of problems where the *preference order* for the adjacent interchange of jobs (the *adjacent interchange order*) is a complete order (i.e. a sequence). *Johnson's rule* establishes such a sequence for the two-machine maximum completion time flow shop problem, while the *rules* of *Smith* and *Jackson* state the optimality of the *weighted shortest processing time* and *earliest due date* sequences for $1 // \sum w_i C_i$ and $1 // L_{\max}$, respectively. (We use the standard notation to describe scheduling problems and we refer the reader to [12] or [16] for any terminology not defined here.) Another classic interchange result due to Emmons [2] provides a *precedence order* (a partial ordering of the jobs that must be obeyed by at least one optimal sequence) for the total tardiness problem $1 // \sum T_i$. In this case, there is a preference order for the interchange of jobs that permits the interchange of any (not necessarily adjacent) pair of jobs in a sequence if they do not appear in preference order. By performing these interchanges in a particular manner, we can restrict our search space to contain only sequences that obey the preference order.

In this paper, we introduce a new technique, a generalization of pairwise interchange that takes into consideration the composition of the intermediate sequence. This yields a preference order which permits the interchange of a pair of jobs *provided* that the intermediate jobs belong to a restricted subset. The traditional methods of pairwise job interchange can be viewed as special cases of this *subset-restricted interchange,* where the subsets are either uniformly the empty set (adjacent interchange) or the entire job set (nonadjacent interchange). In general, an adjacent interchange order is not a complete order and therefore it is *not* a precedence order, as we shall see for the $1 / r / L_{\max}$ problem, using an example due to Lageweg et al. [10]. We prove, however, that using subset-restricted interchange for the class of regular, single machine scheduling problems $1 / r / f_{\max}$, we can derive a precedence order that is a suborder of the adjacent interchange order . In these problems, each job i has an associated nondecreasing, real valued cost function $f_i(t)$, the cost of completing i at time t, and the objective is to minimize the maximum cost $f_{\max}$. The precedence orders we derive have the property that they are defined *independently* of the processing times. This makes them especially useful in applications with stochastic or ill-defined processing times. We use only certain extreme values of the other job parameters that display a 'staircase-like' structure. In addition, the precedence orders derived belong to a special class of partial orders, the interval orders [5]. This also leads to the complexity implication that the above problems are *strongly NP-hard* for interval-ordered tasks. Our results can be viewed as a *unified treatment* of job interchange, that *generalizes* well-known rules for deriving dominance relations and *extends* the pyramid precedence orders of Erschler et al. ([3] and [4]) for $1 / r / L_{\max}$

to general $1/r/f_{\max}$ problems. We generalize the pairwise interchange and insertion operations of Monma [14], and introduce 'pyramid-like' structures of higher dimension than two, extending the 2-dimensional staircases of [3] and [4]. In our unified theory, $1/r/L_{\max}$ represents the most special case in which the adjacent interchange order is a linear order.

The paper is organized as follows. In the next section, we introduce the preliminary definitions and notation for sequencing problems, partial orders, and interchange operators. In section 3, we define the adjacent interchange order $\lessdot$ and the interchange regions for $1/r/f_{\max}$. In section 4, we derive a precedence order $\prec$ for the linearly ordered case of $1/r/f_{\max}$, which includes the $1/r/L_{\max}$, $1\left/r,\overline{d}\right/C_{\max}$, and $1/r/WC_{\max}$ problems. We extend this to the $1/r/WL_{\max}$ and $1/r/f_{\max}$ problems in sections 5 and 6.

2 Preliminary definitions and notation

2.1 Sequencing notation

We call a scheduling problem a sequencing problem if any schedule can be completely specified by the sequence in which jobs are performed. This is the case for non-preemptively scheduling a single machine with a regular performance measure. Let $J = \{1, 2, \ldots, n\}$ be the set of jobs to be sequenced on a single machine. Jobs are characterized by a list of parameters (e.g. for the $1/r/L_{\max}$ problem each job j possesses a release time $r_j \geq 0$, a processing time $p_j \geq 0$, and a due date $d_j \geq 0$, the lateness for job j is $L_j = C_j - d_j$, where C_j is its completion time). A sequence s on J is a function from $\{1, 2, \ldots, n\}$ to J represented by the n-tuple $(s(1), s(2), \ldots, s(n))$, where $s(i)$ is the i^{th} job in sequence s (e.g. for the maximum lateness problem $L_{\max}(s) = \max_i \left(C_{s(i)} - d_{s(i)}\right)$). For the sequencing problems that we study, the adjacent interchange order will be a partial order defined by the parameters of the jobs. Thus we introduce certain definitions for partially ordered sets (posets).

2.2 Partial orders

By a partially ordered set we mean a pair $P = (X, \leq_P)$ consisting of a set X together with a binary relation $\leq_P$ on $X \times X$ which is reflexive, antisymmetric, and transitive. For $u, v \in X$, $u \leq_P v$ is interpreted as u is less than or equal to v in P. Similarly, $u <_P v$ means that $u \leq_P v$ and $u \neq v$. The usual symbols $\leq$ and $<$ will be reserved for relations between real numbers. A partial order $P = (X, \leq_P)$ is a *linear order* (or *complete order*) if for every pair $(u, v) \in X \times X$ either $u \leq_P v$ or $v \leq_P u$. Given a pair of partial orders $P = (X, \leq_P)$ and $Q = (X, \leq_Q)$ on the same set X, we call Q an *extension* of P (P a *suborder* of Q) if $u \leq_P v$ implies $u \leq_Q v$ for all $u, v \in X$. A partial order $Q = (X, \leq_Q)$ is a *linear extension* of a partial order $P = (X, \leq_P)$, if Q is a linear order that extends P. Given two partial orders $P_1 = (X, \leq_{P_1})$ and $P_2 = (X, \leq_{P_2})$, we can define the partial order $P_1 \cap P_2 = (X, \leq_{P_1 \cap P_2})$, the *intersection* of P_1 and P_2, where $u \leq_{P_1 \cap P_2} v$ if and only if $u \leq_{P_1} v$ and $u \leq_{P_2} v$ for all $u, v \in X$. The *dimension* of a partial order $P = (X, \leq_P)$, denoted by $\dim(P)$, is the smallest l such that there exists a set $\{Q_1, Q_2, \ldots, Q_l\}$ of linear extensions of P such that $P = \cap_{i=1}^{l} Q_i$. A subset $I \subseteq X$ is an *ideal* of P if for every $v \in I$ and $u \in X$ such that $u \leq_P v$ we have $u \in I$. Similarly, $F \subseteq X$ is a *filter* of P if for every $u \in F$ and $v \in X$ such that $u \leq_P v$ we have $v \in F$. For every $v \in X$ the *principal ideal* $I(v)$ is defined by $I(v) = \{u \in X \,|u \leq_P v\}$ and the *principal filter* $F(v)$ is defined by $F(v) = \{u \in X \,|v \leq_P u\}$.

2.3 Interchange operators

We follow Monma [14] in defining our interchange operators. Let s_1 be a sequence with job m preceeding job k. In general, s_1 is of the form $s_1 = (AmBkC)$, where A, B and C are subsequences of J. We define three types of interchanges of jobs k and m that leave k preceeding m in the resulting sequence s_2.

1. *Pairwise Interchange(PI)*

   $s_2 = (AkBmC)$

2. *Backward Insertion(BI)*

   $s_2 = (ABkmC)$

3. *Forward Insertion(FI)*

   $s_2 = (AkmBC)$

If we let B be the set of jobs in sequence B (we do not distinguish between these), then each of these interchanges reduces to adjacent pairwise interchange in the case when $B = \emptyset$. This leads to the definition of the *adjacent interchange order.*

Definition 1 *A partial order $\lessdot$ is an* adjacent interchange order *for a sequencing function f if it satisfies the Adjacent Pairwise Interchange Condition: For all jobs k, m and sequences A, C*
$k \lessdot m$ implies $f(AkmC) \leq f(AmkC)$.

Note that all of the above interchanges involve interchanging k, m or both k and m around sequence B. Intuitively, whether or not an interchange leads to a reduction in cost (for a given sequencing function f and adjacent interchange order $\lessdot$), should depend on the composition of B. This involves placing restrictions on certain of the parameters of the jobs in B. In the case when interchangability does not depend on the composition of B, then $\lessdot$ is a *precedence order* for the sequencing problem, i.e. there exists an optimal sequence that is a linear extension of $\lessdot$. Such an example is the precedence order $\lessdot$ defined by

$$k \lessdot m \Leftrightarrow p_k \leq p_m \text{ and } d_k \leq d_m$$

for the total tardiness problem on a single machine $1 // \sum T_i$ [2]. Note that here $\lessdot$ is the intersection of the $\leq_p$ and $\leq_d$ orders. In general, an adjacent interchange order $\lessdot$ is not necessarily a precedence order, as it can be demonstrated by the instance of the maximum lateness problem $1 /r/ L_{\max}$, shown in its equivalent delivery time form in Example 1 [10]. The delivery time version is defined by triples (r_j, p_j, q_j) where $q_j = T - d_j$ is the delivery time for job $j \in J$ and T is a constant chosen so that $T \geq \max\{d_j \,|j \in J\}$. If we define $L_j' = C_j + q_j$, then $L_j' = C_j + T - d_j = L_j + T$ and $L_{\max}' = L_{\max} + T$. As we shall see in the next section, the adjacent interchange order $\lessdot$ for $1 /r/ L_{\max}$ is defined by $k \lessdot m \Leftrightarrow r_k \leq r_m$ and $q_k \geq q_m$. For the 5 job example specified below we have $4 \lessdot 2$, however, the unique optimal sequence is $(1, 2, 3, 4, 5)$ with $L_{\max}' = 11$. Thus $\lessdot$ is *not* a precedence order, since the unique optimal sequence is not a linear extension of $\lessdot$.

Example 1 *A 5 job problem to illustrate that $\lessdot$ is not necessarily a precedence order.*

j	1	2	3	4	5
r_j	0	2	3	0	7
p_j	2	1	2	2	2
q_j	5	2	6	3	2

We consider interchanges that are restricted by conditions on B and define the subset-restricted interchange conditions as follows.

Definition 2 *An adjacent interchange order $\lessdot$ together with the collection of subsets $R^{PI} = \left\{R^{PI}_{k \lessdot m} \,|k \lessdot m\right\}$ satisfies the Restricted Pairwise Interchange Condition for a sequencing function f if*
for all jobs k, m and sequences A, B, C
$k \lessdot m$ and $B \subseteq R^{PI}_{k \lessdot m}$ imply $f(AkBmC) \leq f(AmBkC)$.

Definition 3 *An adjacent interchange order $\lessdot$ together with the collection of subsets $R^{BI} = \left\{ R^{BI}_{k \lessdot m} \,|k \lessdot m \right\}$ satisfies the Restricted Backward Insertion Condition for a sequencing function f if*
for all jobs k, m and sequences A, B, C
$k \lessdot m$ and $B \subseteq R^{BI}_{k \lessdot m}$ imply $f\,(ABkmC) \leq f\,(AmBkC)$.

Definition 4 *An adjacent interchange order $\lessdot$ together with the collection of subsets $R^{FI} = \left\{ R^{FI}_{k \lessdot m} \,|k \lessdot m \right\}$ satisfies the Restricted Forward Insertion Condition for a sequencing function f if*
for all jobs k, m and sequences A, B, C
$k \lessdot m$ and $B \subseteq R^{FI}_{k \lessdot m}$ imply $f\,(AkmBC) \leq f\,(AmBkC)$.

In the following sections, we show how to use subset-restricted interchange and the above three conditions to derive a precedence order $\prec$ on the jobs. This precedence order $\prec$ is always a suborder of the adjacent interchange order $\lessdot$.

3 Interchange regions

In this section, we derive the interchange regions (subsets) for the general problem $1\,/r\,/f_{\max}$. In this problem, each job j has an associated *nondecreasing*, real valued cost function f_j, where $f_j\,(t)$ is the cost of completing job j at time t, and the objective is to minimize $f_{\max} = \max\limits_{1 \leq j \leq n} f_j\,(C_j)$ over all sequences. We order the jobs according to $\geq_f$, where $f_i \geq_f f_j \Leftrightarrow f_i\,(t) \geq f_j\,(t)$ *for all* $t \geq 0$. Note that, in the general case, $\geq_f$ does not order every pair i and j, it may be only a partial order. Two special, linearly ordered cases of $\geq_f$ occur for the lateness objective, where $f_j\,(t) = t + q_j$, and the weighted completion time objective, where $f_j\,(t) = w_j t$. Hall [7] considered the $\geq_f$ order and noticed that the linear ordering property makes it possible to extend Potts' [17] approximation algorithm for the $1\,/r\,/L_{\max}$ problem to the $1\,/r\,/f_{\max}$ problem *when* $\geq_f$ is a linear order.

The adjacent interchange order and the restricted subsets for $1\,/r\,/f_{\max}$ are defined below. We note that these definitions use *no processing time information.* This means that all of the subsequent results are true *irrespective* of job processing times.

Definition 5 *Adjacent interchange order: $k \lessdot m \Leftrightarrow r_k \leq r_m$ and $f_k \geq_f f_m$.*

Note that this order is not complete, but it satisfies the Adjacent Pairwise Interchange Condition.

Definition 6 *Given $k \lessdot m$, define the following* **subsets** *of jobs:*
(i) $R^{PI}_{k \lessdot m} = \{j \,|r_j \le r_m, f_k \ge_f f_j\}$
(ii) $R^{BI}_{k \lessdot m} = \{j \,|r_j \le r_m\}$
(iii) $R^{FI}_{k \lessdot m} = \{j \,|f_k \ge_f f_j\}$.

Theorem 1 $\lessdot$ *together with the collection of subsets* $R^{PI} = \left\{R^{PI}_{k \lessdot m} \,|k \lessdot m\right\}$ *satisfies the Restricted Pairwise Interchange Condition for* $1\,/r\,/f_{\max}$.

Proof. Given a sequence s, recall that $f_{\max}(s) = \max\limits_{1 \le j \le n} f_{s(j)}\left(C_{s(j)}\right)$, where $C_{s(j)}$ is the completion time of job $s(j)$. We construct a directed graph $G(s)$ to evaluate $f_{\max}(s)$ (see Figure 1). From the source node 0 of $G(s)$ there is a directed edge of length $r_{s(j)}$ to each job node $s(j)$ $(j = 1, 2, \ldots, n)$, and between each pair of jobs $s(j)$ and $s(j+1)$ there is a directed edge of length $p_{s(j)}$. $G(s)$ has the property that the start time of job $s(j)$ in sequence s is the length of the longest path from 0 to $s(j)$, and to obtain $C_{s(j)}$ we add $p_{s(j)}$ to the length of this path. We represent paths from 0 to $s(j)$ by pairs $(s(i), s(j))$, $1 \le i \le j \le n$, where $s(i)$ is the endpoint of the first arc, $(0, s(i))$, of the path. Then by definition

$$f_{\max}(s) = \max_{1 \le j \le n}\left[f_{s(j)}\left(\max_{\substack{(s(i),s(j))\\ 1 \le i \le j}}\left(r_{s(i)} + \sum_{l=i}^{j} p_{s(l)}\right)\right)\right] = \max_{\substack{(s(i),s(j))\\ 1 \le i \le j \le n}}\left[f_{s(j)}\left(r_{s(i)} + \sum_{l=i}^{j} p_{s(l)}\right)\right]$$

and we can evaluate $f_{\max}(s)$ as the maximum over all such pairs $(s(i), s(j))$ in $G(s)$.

Let $s_1 = (AmBkC)$ be a sequence with the property that $k \lessdot m$ and $B \subseteq R^{PI}_{k \lessdot m}$. We apply pairwise interchange to s_1 and obtain sequence $s_2 = (AkBmC)$. We demonstrate that s_2 is not worse than s_1 by showing that for *every* pair of jobs in s_2 there exists a *dominating pair* in s_1 with a not smaller f value. For example, consider pair (k, m) in s_2, then it has the dominating pair (m, k) in s_1 (see Figure 2): That is,

$$f_m\left(r_k + p_k + \sum_{b \in B} p_b + p_m\right) \le f_k\left(r_m + p_m + \sum_{b \in B} p_b + p_k\right),$$

which holds since $k \lessdot m$ implies $r_k \le r_m$ and $f_k(t) \ge f_m(t)$ *for all* t, from which it follows that $f_m\left(r_k + p_k + \sum_{b \in B} p_b + p_m\right) \le f_k\left(r_k + p_m + \sum_{b \in B} p_b + p_k\right)$, and finally f_k is *nondecreasing.*

The following table gives a dominating pair in s_1 for each pair in s_2. (We use lower case letters a,b or c to refer to arbitrary generic elements of the subsequences A,B or C, respectively). The last three columns contain the arguments why they are dominating pairs.

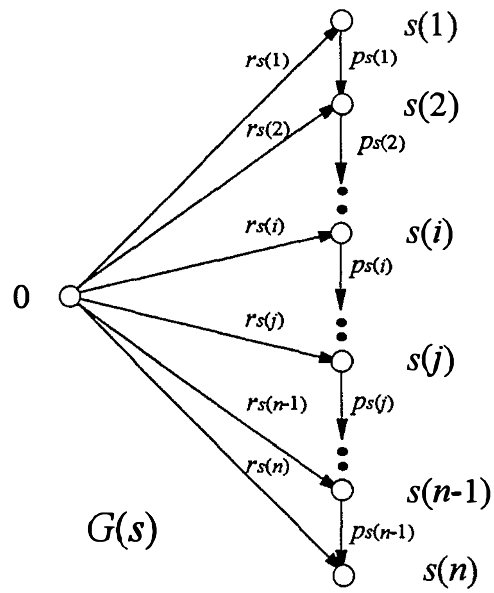

Figure 1: Directed graph $G(s)$ for sequence s.

s_2	s_1			
$(AkBmC)$	$(AmBkC)$	*proof*		
(a,k)	(a,k)			f_k *nondecreasing*
(a,b)	(a,k)		$f_k \geq_f f_b$	f_k *nondecreasing*
(a,m)	(a,k)		$f_k \geq_f f_m$	f_k *nondecreasing*
(a,c)	(a,c)			
(k,b)	(m,k)	$r_k \leq r_m$	$f_k \geq_f f_b$	f_k *nondecreasing*
(k,m)	(m,k)	$r_k \leq r_m$	$f_k \geq_f f_m$	f_k *nondecreasing*
(k,c)	(m,c)	$r_k \leq r_m$		f_c *nondecreasing*
(b,m)	(m,k)	$r_b \leq r_m$	$f_k \geq_f f_m$	f_k *nondecreasing*
(b,c)	(m,c)	$r_b \leq r_m$		f_c *nondecreasing*
(m,c)	(m,c)			f_c *nondecreasing*

■

Theorem 2 *$<$ together with the collection of subsets $R^{BI} = \left\{ R^{BI}_{k<m} \, | k<m \right\}$ satisfies the Restricted Backward Insertion Condition for $1 \, / r \, / f_{\max}$.*

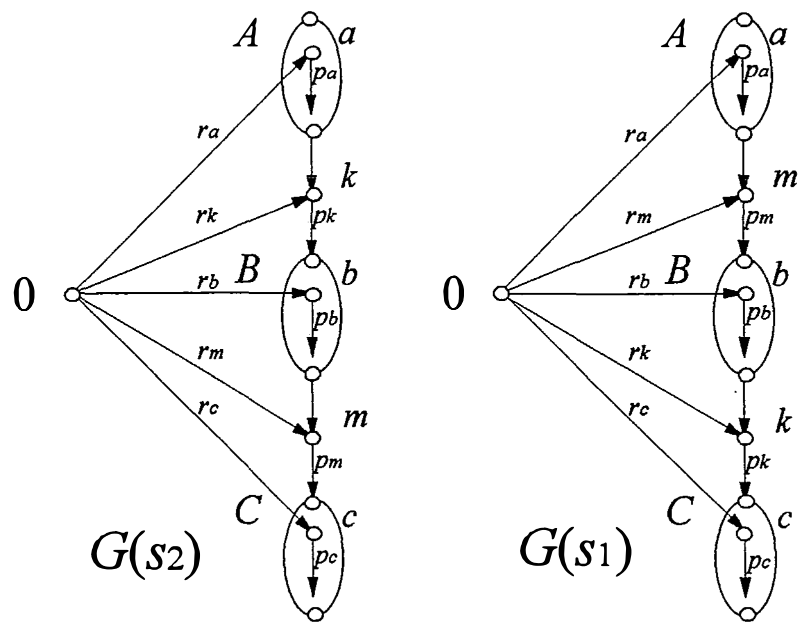

Figure 2: Directed graphs $G(s_2)$ and $G(s_1)$.

Proof. The proof is totally analogous to that for pairwise interchange. The following table gives the corresponding dominating pairs.

s_2	s_1			
$(ABkmC)$	$(AmBkC)$	*proof*		
(a,b)	(a,b)			f_b *nondecreasing*
(a,k)	(a,k)			f_k *nondecreasing*
(a,m)	(a,k)		$f_k \geq_f f_m$	f_k *nondecreasing*
(a,c)	(a,c)			
(b,k)	(b,k)			
(b,m)	(m,k)	$r_b \leq r_m$	$f_k \geq_f f_m$	f_k *nondecreasing*
(b,c)	(m,c)	$r_b \leq r_m$		f_c *nondecreasing*
(k,m)	(m,k)	$r_k \leq r_m$	$f_k \geq_f f_m$	f_k *nondecreasing*
(k,c)	(m,c)	$r_k \leq r_m$		f_c *nondecreasing*
(m,c)	(m,c)			f_c *nondecreasing*

■

Theorem 3 $\lessdot$ *together with the collection of subsets* $R^{FI} = \left\{ R^{FI}_{k \lessdot m} \mid k \lessdot m \right\}$ *satisfies the Restricted Forward Insertion Condition for* $1 / r / f_{\max}$.

Proof. The proof is totally analogous to that for pairwise interchange. The following table gives the corresponding dominating pairs.

s_2	s_1			
$(AkmBC)$	$(AmBkC)$	*proof*		
(a,k)	(a,k)			f_k *nondecreasing*
(a,m)	(a,k)		$f_k \geq_f f_m$	f_k *nondecreasing*
(a,b)	(a,k)		$f_k \geq_f f_b$	f_k *nondecreasing*
(a,c)	(a,c)			
(k,m)	(m,k)	$r_k \leq r_m$	$f_k \geq_f f_m$	f_k *nondecreasing*
(k,b)	(m,k)	$r_k \leq r_m$	$f_k \geq_f f_b$	f_k *nondecreasing*
(k,c)	(m,c)	$r_k \leq r_m$		f_c *nondecreasing*
(m,b)	(m,b)			
(m,c)	(m,c)			f_c *nondecreasing*
(b,c)	(b,c)			f_c *nondecreasing*

■

Remark 1 We observe that for any $k \lessdot m$, we have $R^{PI}_{k\lessdot m} = R^{BI}_{k\lessdot m} \cap R^{FI}_{k\lessdot m}$. Thus if $B \subseteq R^{PI}_{k\lessdot m}$, then this implies not only $f(AkBmC) \leq f(AmBkC)$, but also $f(ABkmC) \leq f(AmBkC)$ and $f(AkmBC) \leq f(AmBkC)$.

The preceding theorems could directly be used in branch-and-bound algorithms for restricting the search space on sequences. This, however, would require branching on sequences and storing for all pairs $k \lessdot m$ the subsets of Definition 6 and the testing of membership in these, which would be time consuming and inefficient. In the following sections, we show that there is a much more effective way to restrict the search space, by proving that there is a precedence order on the jobs.

We also note that by simply modifying release times and processing times, problems in which the jobs have setup times can be handled as well. Two forms of job setups can be considered: either a setup s_i is *attached* to job i and it cannot be performed before r_i, or it is *detached* and it can be performed prior to r_i, while the machine is idle and waiting to process job i. Detached setups can be dealt with by using modified processing times $p_i^{'} = p_i + s_i$, while attached setups can be handled using modified release times $r_i^{'} = \max\{0, r_i - s_i\}$ and processing times $p_i^{'} = p_i + s_i$. Thus, our theory of precedence constraints applies to the case with setups too.

4 The linearly ordered case

In this section we use subset-restricted interchange to derive a precedence order $\prec$, for the case when $\geq_f$ is a linear order. This means that for *each* pair of jobs i and j either $f_i(t) \geq f_j(t)$ *for all* t, or $f_j(t) \geq f_i(t)$ *for all* t. That is, we can *completely* arrange the jobs in nonincreasing f order according to $\geq_f$.

For the linearly ordered case, we are able to represent the adjacent interchange order in the plane using the r and f orders as the x and y axes respectively. Here jobs are represented by points with preferences toward the origin, i.e. $k \lessdot m$ if k is closer to the origin than m in both the r and f orders. The principal ideals and filters are represented by quadrants through these points. That is, let job i be represented by the point (r_i, f_i). If we divide the plane into quadrants using the lines $r = r_i$ and $f = f_i$, then the SW and NE quadrants correspond to $I(i)$ and $F(i)$, the principal ideal and filter for job i (see Figure 3).

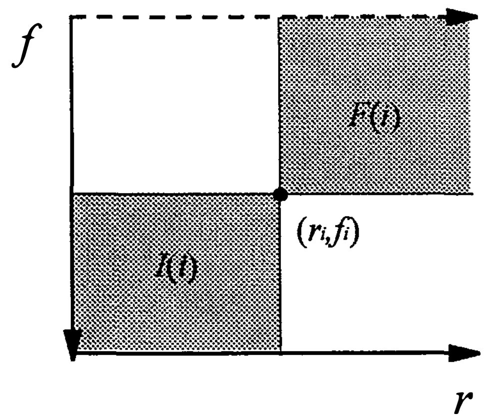

Figure 3: Ideals and filters for $\lessdot$.

This planar represention was used by Merce [13] to derive a precedence order for the problem of minimizing the makespan in the presence of release times and deadlines $(1 / r, \bar{d} / C_{\max})$. Fontan [6] noted that if we consider due dates instead of deadlines then the same order is a precedence order for the lateness model $1 / r / L_{\max}$. They do not consider the adjacent interchange order explicitly ([3] and [4]), rather they define their order using certain extreme points in the plane, called summits. These *summits*, represented by S_i ($i = 1$ to N), are the jobs that form a staircase boundary in the plane and satisfy the property that their SE quadrant minus the boundary is empty. The summits are completely ordered $S_1 \lessdot S_2 \lessdot \ldots \lessdot S_N$. For each summit S_i, they define a *pyramid* $P(S_i)$, which is its NW quadrant without its boundary lines but including S_i. Jobs are classified using pairs that represent their membership in pyramids. For $j \in J$, they define $u(j) = \min\{i \,|\, j \in P(S_i)\}$, and $v(j) = \max\{i \,|\, j \in P(S_i)\}$. With these, they define the partial order $\prec$ by $k \prec m \Leftrightarrow v(k) < u(m)$, and show that $\prec$ is a precedence order. The planar representation for the 5 job problem of Example 1 is shown in Figure 4: Job 3 and 5 are the summits S_1 and S_2, respectively. $P(S_1) = \{1,2,3,4\}$ and $P(S_2) = \{5\}$. We have $v(i) = 1$ for $i = 1,2,3,4$ and $u(5) = 2$, which implies by definition of $\prec$ that jobs 1,2,3 and 4 must precede job 5. Notice that the unique optimal sequence $(1,2,3,4,5)$ is a linear extension of $\prec$ (as we would expect), but not a linear extension of $\lessdot$ (as we saw earlier).

Example 2 is another instance of $1\,/r/\,L'_{\max}$, this time with $N = 9$ summits. For this instance, the optimal $L'_{\max} = 114$ and the sequence $(1,S_1,k,S_2,S_3,S_4,S_5,3,S_6,S_7,$ $m,2,S_8,S_9,4)$ is an optimal sequence, which is also a linear extension of $\prec$. To further illustrate how $\prec$ is defined, consider jobs k and m in Figure 5: Here $k \prec m$, since $v(k) = 5 < 6 = u(m)$. (The points D_i and M_i in brackets are used in our unified theory and will be explained later in this section.)

Example 2 *A 15 job instance of* $1\,/r/\,L'_{\max}$ *with* $N = 9$ *summits, to further illustrate* $\prec$.

j	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	k	m	1	2	3	4
r_j	15	26	34	40	48	57	65	65	73	19	51	8	22	30	38
p_j	6	8	8	7	6	10	2	5	7	5	3	5	8	8	10
q_j	53	46	40	36	36	27	19	13	9	32	17	43	16	22	6

This pyramid-based precedence order $\prec$ very heavily uses the planar representation. It can be reinterpreted in partial order terminology, however, which will allow us to extend these precedence constraints to other, more general cost functions and higher dimensions. The summits S_i $(i = 1,2,\ldots,N)$ are *maximal elements* of a related partial order $\lessdot^c$, which is defined by $k \lessdot^c m \Leftrightarrow r_k < r_m$ and $f_k <_f f_m$, the conjugate of $\lessdot$. Two partial orders on the same set are *conjugate* if every pair of distinct elements is comparable in exactly one of these partial orders. This is clearly the case if we compare the principal ideals and filters for $\lessdot$ and $\lessdot^c$, using the planar representation. For $\lessdot$, these are the SW and NE quadrants, respectively, with the boundary lines included. For $\lessdot^c$, these are the NW and SE quadrants minus the boundary lines (compare Figures 3 and 6). By a well known theorem of Dushnik and Miller [1] from partial order theory, $\lessdot^c$ exixts if and only if $\dim(\lessdot) \leq 2$. In this context, the pyramids are just the *principal ideals* of $\lessdot^c$. The $u(j)$ and $v(j)$ defined in

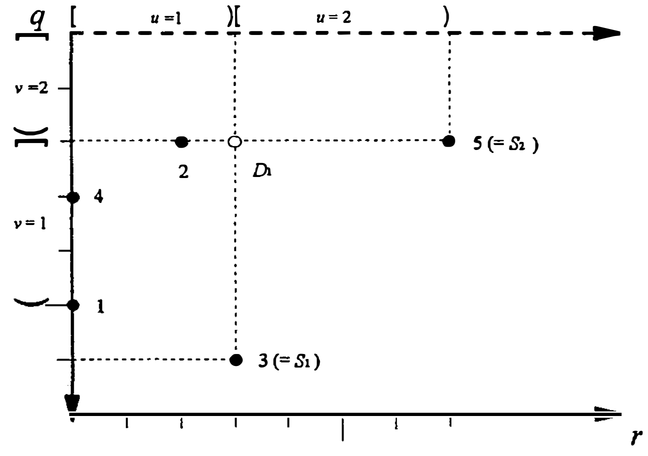

Figure 4: Planar representation for 5 job problem.

[4] implicitly use the fact that $\lessdot^c$ has an *interval containment order* representation, i.e. there exist intervals $\{I_j | j \in J\}$ such that $i \lessdot^c j$ iff $I_i \subset I_j$ for $i, j \in J$. In their representation of $1/r/L_{\max}$, these intervals are just $I_j = [r_j, d_j]$. On the other hand, by the same theorem of Dushnik and Miller [1], a poset has an interval containment representation iff its dimension is 2. Thus, the $u(j)$ and $v(j)$ can be defined for *any* problem for which $\dim(\lessdot) \leq 2$, but it can be defined *only* for such problems. Of course, $\dim(\lessdot) \leq 2$ is equivalent to $>_f$ being a linear order, so the $u(j)$ and $v(j)$ can be defined *only* in this case.

We consider an *alternate representation* for $\prec$, using the set of corner point boundary jobs $M = \{M_i \,| i = 1, 2, \ldots, H+1\}$, and the set of points on their inscribed diagonal $\Delta = \{D_1, D_2, \ldots, D_H\}$ (see Figure 5). The set $M \subseteq S$ is the subset of boundary jobs with empty SE quadrants, and we call Δ the set of *diagonal* points. (The set Δ is defined more precisely by a recursive algorithm below.) Note that the points in Δ may represent fictitious jobs. We augment the partial order $P = (J, \lessdot)$ by these diagonal points and call it $P_\Delta = (J_\Delta, \lessdot_\Delta)$, where $J_\Delta = J \cup \Delta$ and $\lessdot_\Delta$ is the planar order with these diagonal points included. Jobs $k \lessdot m$ ($k,m \in J$) are *separated by* Δ (are Δ*-separated)* if there exists a D_i ($i = 1, 2, \ldots, H$) such that $k \in I(D_i)$ and $m \in F(D_i)$. Notice that $v(k) < u(m)$ when k and m are Δ-separated. Δ induces a *partition* of P into separable and nonseparable pairs that can be used to define $\prec$: It can be easily verified for $k \lessdot m$ ($k,m \in J$) that $k \prec m$ if and only if k and m are

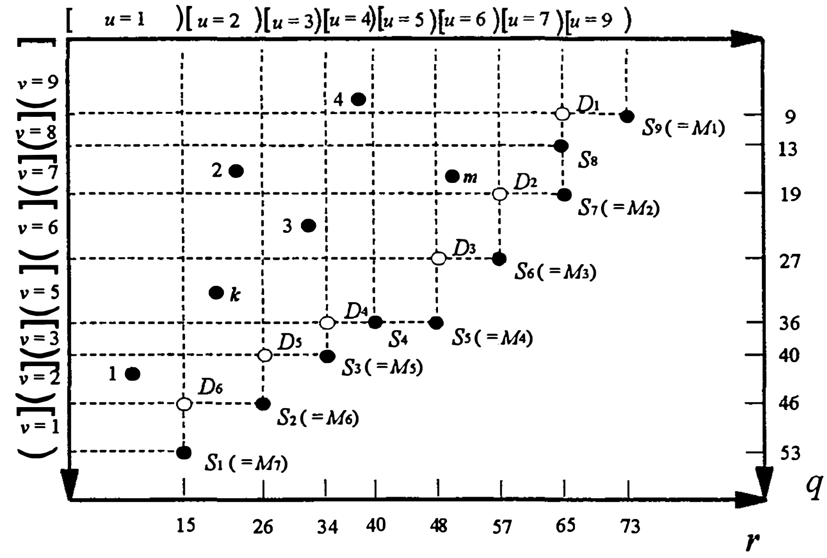

Figure 5: Planar representation for the 15 job problem.

separated by Δ. As an example of this representation, consider again Example 1 in Figure 4: We see that diagonal point D_1 separates job 5 and jobs 1,2,3 and 4, and this is the *only* separation present. Thus, by this Δ-separation representation of $\prec$, jobs 1,2,3 and 4 must precede job 5. In comparison to the pyramid representation, this new representation for $\prec$ has the advantage that it does *not* require that $\lessdot$ possess a conjugate $\lessdot^c$, and thus is *not restricted* by the dimension of $\geq_f$. This will enable us to generalize the whole theory to the nonlinearly ordered, higher dimensional case in later sections.

The diagonal Δ can be obtained from the set M of corner point boundary jobs using the following simple greedy procedure. Assume that there are K distinct r values denoted by r^i for $i = 1, 2, \ldots, K$, and $r^1 > r^2 > \cdots > r^K$. Define the function $f^i = \max\{f_j \,|r_j = r^i\}$ with values $f^i(t) = \max\{f_j(t)\,|r_j = r^i\}$, this is the *maximum,* according to $\leq_f$, of the jobs on level r^i. Note that by the linearity condition on $\geq_f$, we have that $f^i = f_j$ for some job j with $r_j = r^i$ (e.g. for $WC_{\max}$ $f^i = w^i t$, where $w^i = \max\{w_j\,|r_j = r^i\}$; and for $L_{\max}$ $f^i = t - d^i$, where $d^i = \min\{d_j\,|r_j = r^i\}$).

Algorithm Δ for linearly ordered $\geq_f$

Let $(r_{M_1}, f_{M_1}) = (r^1, f^1)\,, l = 1$

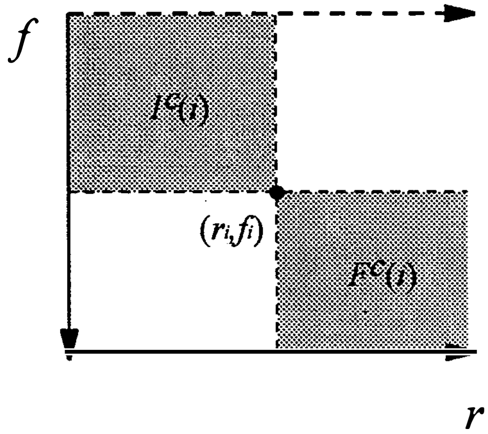

Figure 6: Ideals and filters for $<^c$.

For $i = 2$ to K

If $f_{M_l} \geq_f f^i$ then go to Next i

Else

$(r_{M_l+1}, f_{M_l+1}) = (r^i, \max\{f^i, f_{M_l}\})$ /* This defines M_{l+1}

$(r_{D_l}, f_{D_l}) = (r^i, f_{M_l})$ /* This defines D_l

Increase l to $l+1$

Next i

The procedure looks at the r-levels r^i $(i = 1, 2, \ldots, K)$ and compares the largest f on this level (f^i) with the largest f obtained so far (f_{M_l}). If f^i represents a strict increase compared to f_{M_l}, the point (r^i, f^i) becomes the new corner point (r_{M_l+1}, f_{M_l+1}). Next, we prove that $\prec$, defined by $k \prec m$ if and only if k and m are Δ-separated, is a precedence order for *all* sequencing problems for which $\geq_f$ is a linear order. This *unifies* and *generalizes* the results in ([3] and [4]).

Theorem 4 *$\prec$ is a precedence order for $1/r/f_{\max}$, if $\geq_f$ is a linear order.*

Proof. We use subset-restricted interchange in the proof. The following observations immediately follow from the construction of Δ:

$$\begin{array}{ccccccc} D_1 & \succ & D_2 & \succ & \cdots & \succ & D_H \\ F(D_1) & \subset & F(D_2) & \subset & \cdots & \subset & F(D_H) \\ I(D_1) & \supset & I(D_2) & \supset & \cdots & \supset & I(D_H) \end{array} .$$

Furthermore, for all $b \in J \setminus F(D_i)$ and $m \in F(D_i)$ we have $r_b \leq r_{D_i} \leq r_m$ for any i. This the *crucial* property used throughout our proof. Let s be any optimal sequence. If every job in $I(D_1)$ is before every job in $F(D_1)$, then all jobs separated by D_1 are already in $\prec$ order, and consider $I(D_2)$ and $F(D_2)$. Otherwise, let $k_1 \in I(D_1)$ be the last job in s that is after some job from $F(D_1)$, and let m_1 be the last such job from $F(D_1)$ before k_1. That is $s = (A_1 m_1 B_1 k_1 C_1)$, where $C_1 \cap I(D_1) = \emptyset$ and $B_1 \subset J \setminus F(D_1)$. By the above property, we have $r_{b_1} \leq r_{D_1} \leq r_{m_1}$ for all $b_1 \in B_1$, which implies that $B_1 \subseteq R^{BI}_{k_1 \lessdot m_1}$. Thus, by subset-restricted interchange, we can insert m_1 *backward* just after k_1 to obtain the alternative optimal sequence $(A_1 B_1 k_1 m_1 C_1)$. Following in this way, inserting the last job in $F(D_1)$ *backward* after k_1 until there are no such jobs, we obtain sequence s_1 which is an optimal sequence with the property that $I(D_1)$ is before $F(D_1)$. Continuing similarly, if $I(D_2)$ is before $F(D_2)$ in s_1, then all jobs separated by D_2 are already in $\prec$ order, and consider $I(D_3)$ and $F(D_3)$. Otherwise, let $k_2 \in I(D_2)$ be the last job in s_1 after some job from $F(D_2) \setminus F(D_1)$ (since $I(D_2) \subset I(D_1)$ and $I(D_1)$ is before $F(D_1)$ in s_1), and let m_2 be the last such job from $F(D_2) \setminus F(D_1)$. Similarly, we have $s_1 = (A_2 m_2 B_2 k_2 C_2)$, where $C_2 \cap I(D_2) = \emptyset$ and $B_2 \subseteq J \setminus F(D_2)$. As above, we have that $r_{b_2} \leq r_{D_2} \leq r_{m_2}$ for all $b_2 \in B_2$, which implies that $B_2 \subseteq R^{BI}_{k_2 \lessdot m_2}$. Thus we can insert m_2 *backward* just after k_2 to obtain the sequence $(A_2 B_2 k_2 m_2 C_2)$. When all such jobs in $F(D_2) \setminus F(D_1)$ have been inserted after k_2, we obtain sequence s_2, an optimal sequence with the property that $I(D_2)$ is before $F(D_2)$ and $I(D_1)$ is before $F(D_1)$. Continuing similarly, we obtain s_i for $i = 3, 4, \ldots, H$. Then s_H is an optimal sequence with the property that $I(D_i)$ is before $F(D_i)$ for $i = 1, 2, \ldots, H$. Thus s_H is a linear extension of $\prec$, and we have that $\prec$ is a precedence order indeed. ■

The original proof of Erschler et al. [4], for $1/r/L_{\max}$, used *pairwise interchange.* This proof can also be modified to carry over to other cost functions f *when* $\geq_f$ is a linear order. We chose to present a proof using backward insertion, however, because this extends to the (nonlinearly ordered) general case. A proof using forward insertion can also be obtained by proceeding in the opposite direction. This is due to the duality of the operations and regions for the linearly ordered case. Interestingly, however, the proof based on forward insertion is not extendable to the general case either, as the duality of regions no longer holds.

It is well known that the main source of difficulty in all $1/r/f_{\max}$ problems is the fact that at any time the machine becomes available, it may be better to wait for a yet unreleased job rather than to schedule one of the jobs available. The partition of P by Δ means that the *only* jobs for which it may be worth waiting are the ones which are *not* separated by Δ from the currently available jobs.

Although Theorem 4 requires the linearity of $\geq_f$, it covers a number of well-studied scheduling problems. In addition to the ones studied in ([3] and [4]), we mention one as an example.

Corollary 1 *$\prec$ is a precedence order for $1/r/WC_{\max}$ and, in this case, $\geq_f$ is the order which orders the jobs in nonincreasing w order.*

Theorem 4 also has interesting complexity implications. As we have discussed earlier, the set $S = \{S_1, S_2, \ldots, S_N\}$ is linearly ordered by $\lessdot$, and we can reinterpret the pairs $(u(j), v(j))$ as intervals $I_j^{'} = [u(j), v(j)]$ in S. As $\prec$ was originally defined by $k \prec m \Leftrightarrow v(k) < u(m)$ ([3] and [4]), this gives an *equivalent* interval representation for $\prec$: $k \prec m \Leftrightarrow$ the interval $I_k^{'} = [u(k), v(k)]$ lies strictly to the left of $I_m^{'} = [u(m), v(m)]$ in the complete order $(S, \lessdot)$. Partial orders $(P, \leq_P)$ that possess such an interval representation on a linearly ordered set (where $x \leq_P y \Leftrightarrow I_x^{'}$ lies to the left of $I_y^{'}$) are called *interval orders* [5]. This yields the following tightening of previously known complexity results ([7], [12]):

Corollary 2 *$1/r, prec/L_{\max}$ and $1/r, prec/WC_{\max}$ remains NP-hard in the strong sense even with interval order precedence constraints.*

Corollary 2 is interesting, as interval orders have a very special restricted structure [5], but this does not seem to help in reducing the complexity of the scheduling problems mentioned. This is in contrast with the result of [15] which shows that $Pm/p_j = 1, prec/C_{\max}$ is polynomially solvable *for* interval-ordered precedence constraints.

Recall that when the adjacent interchange order $\lessdot$ *itself* is a linear order, it defines an optimal sequence. It can easily be seen, that $\prec$ is equivalent to $\lessdot$ in this case, and so it also defines an optimal sequence. This means that $\prec$ also solves some well known special cases solved by Jackson's rule [8] or Lawler's method [11]: For $1/r/L_{\max}^{'}$, $f_i(t) = t + q_i$ and the jobs are linearly ordered by $f_i \geq_f f_j \Leftrightarrow q_i \geq q_j$. So both $\lessdot$ and $\prec$ are linear orders and define an optimal sequence, for example, when the f order is the same as the r order (the agreeably ordered case of $1/r/L_{\max}^{'}$ in which $r_i \leq r_j \Leftrightarrow q_i \geq q_j$); or one of the orders is trivial (i.e. the f order is linear and *all* the jobs have the same release time $(1//L_{\max})$; or *all* the jobs have the same cost function $(1/r, d_j = d/L_{\max})$. Similar comments apply to the corresponding special cases of $1/r/WC_{\max}$ and $1/r/f_{\max}$.

5 Weighted maximum lateness

We derive first the Δ boundary for the weighted maximum lateness problem $1/r/WL_{\max}$. The $\geq_f$ order is *no longer linear* for this problem and this will also motivate our definition of Δ for the general case.

For the weighted maximum lateness problem, $f_j(t) = w_j(t - d_j)$ for all $j \in J$. Let $g_j(t) = f_j(t) + L$, where the constant L is chosen so that $L \geq \max\{w_j d_j \mid j \in J\}$. That is, $g_j(t) = w_j t + q_j$, where $q_j = L - w_j d_j \geq 0$. We see that $f_k \geq_f f_m \iff (w_k \geq w_m)$ and $(q_k \geq q_m)$. Thus, we can represent the adjacent interchange order $\lessdot$ and the interchange regions using this 2-*dimensional* representation of $\geq_f$. This means that our general definitions from before reduce to the following for $1/r/WL_{\max}$.

$$
\begin{aligned}
k \lessdot m \quad & \Leftrightarrow (r_k \leq r_m) \text{ and } (w_k \geq w_m) \text{ and } (q_k \geq q_m) \\
R^{PI}_{k \lessdot m} \quad & = \{j \mid r_j \leq r_m, w_j \leq w_k, q_j \leq q_k\} \\
R^{FI}_{k \lessdot m} \quad & = \{j \mid w_j \leq w_k, q_j \leq q_k\} \\
R^{BI}_{k \lessdot m} \quad & = \{j \mid r_j \leq r_m\}
\end{aligned}
$$

We derive the Δ boundary by modifying the greedy procedure presented earlier. Once again, we assume that there are K distinct r values $r^1 > r^2 > \cdots > r^K$ and let $w^i = \max\{w_j \mid r_j = r^i\}$ and $q^i = \max\{q_j \mid r_j = r^i\}$ for $i = 1, 2, \ldots, K$. We use the same notation as before for the boundary points M_i and the diagonal points D_i. Note that in this case both the M_i and D_i may belong to fictitious jobs.

Algorithm Δ for $1/r/WL_{\max}$

Let $(r_{M_1}, q_{M_1}, w_{M_1}) = (r^1, q^1, w^1), l = 1$

For $i = 2$ to K

If $q_{M_l} \geq q^i$ and $w_{M_l} \geq w^i$ then go to Next i /*i.e. $(q_{M_l}, w_{M_l}) \geq_f (q^i, w^i)$

Else

$(r_{M_{l+1}}, q_{M_{l+1}}, w_{M_{l+1}}) = (r^i, \max\{q^i, q_{M_l}\}, \max\{w^i, w_{M_l}\})$ /* This defines M_{l+1}

$(r_{D_l}, q_{D_l}, w_{D_l}) = (r^i, q_{M_l}, w_{M_l})$ /* This defines D_l

Increase l to $l+1$

Next i.

If we represent $\lessdot$ in 3-dimensions with q, w, and r as the x, y, and z axes, respectively, then (q^i, w^i) is the *least upper bound* according to $\leq_f$ of the jobs on the plane $r = r^i$. (This is well defined by the finiteness of J.) Symbolically, $(q^i, w^i) = \max_2\{(q_j, w_j) \mid r_j = r^i\}$, where we define $\max_2\{(q_j, w_j) \mid j \in I\} = (\max_{j \in I} q_j, \max_{j \in I} w_j)$.

Taking least upper bounds, we greedily construct the sets of —possibly fictitious— jobs Δ and M. These jobs are on the boundary of a *step pyramid,* which contains all of the original jobs inside or on its surface (see Figure 7 for an example with $H = 6$). The definition of $\prec$ is analogous to our new definition of it for the linearly ordered case: $k \prec m$ if k and m are Δ-separated. In the next theorem we prove that $\prec$ is a precedence order for $1\,/r\,/W L_{\max}$.

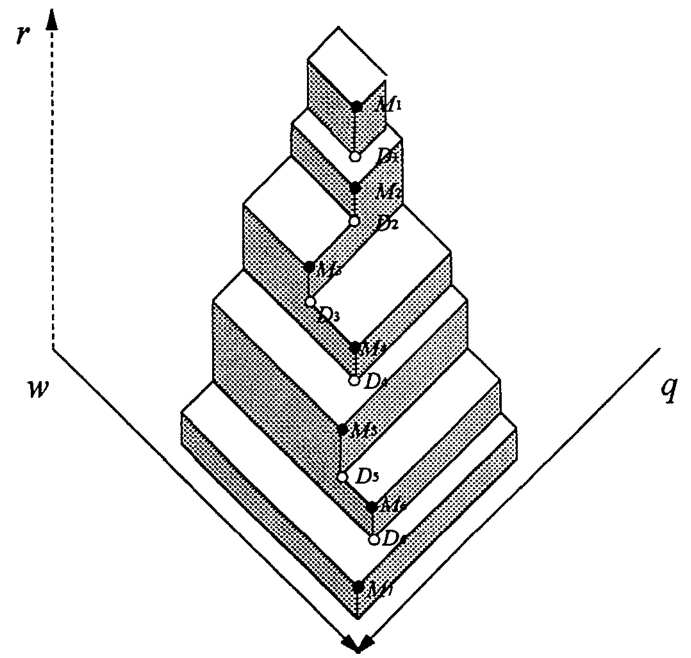

Figure 7: Staircase structure for weighted maximum lateness.

Theorem 5 *$\prec$ is a precedence order for* $1\,/r\,/W L_{\max}$.

Proof. The proof is analogous to that for the linearly ordered case, using backward insertion and the present diagonal Δ. Recall that that proof requires the *critical property* that $r_b \leq r_{D_i}$ for all $b \in J \setminus F(D_i)$ and any i. This is true here because of the greedy way that we construct M: The procedure looks at the r-levels r^i ($i = 1, 2, \ldots, K$) and takes the maximum (in $\leq_f$) of the least upper bound for r^i and the maximum so far. M_i ($i = 1, 2, \ldots, H + 1$) is the i^{th} proper maximum obtained in this way, and D_i ($i = 1, 2, \ldots, H$) is its *projection* onto the r-level of M_{i+1}. By recalling that $\lessdot\, = \leq_r \cap \geq_f$ (Definition 5), we see that D_i is a lower bound according to $\lessdot$ for *all* jobs on higher r-levels, and the critical property holds indeed. The rest of the proof is the same as in the linearly ordered case. ■

In contrast with the case when $\geq_f$ was linearly ordered, the dual proof using Foreward Insertion does *not* follow because the dual condition $f_b \geq_f f_{D_i} \Leftrightarrow (q_b \leq q_{D_i})$ and $(w_b \leq w_{D_i})$ for all $b \in J \setminus I(D_i)$ no longer holds.

6 The general case

For the general case of $1/r/f_{\max}$, recall that $f_k \geq_f f_m \iff f_k(t) \geq f_m(t)$ *for all* t. Similarly to the just discussed 2-dimensional case, we introduce the *least upper bound* in $\leq_f$ for a finite set of jobs $I \subseteq J$ as the nondecreasing function $\operatorname*{pmax}_{j\in I} f_j$ defined as the *pointwise maximum* of the functions f_j, i.e. with values $\left(\operatorname*{pmax}_{j\in I} f_j\right)(t) = \max_{j\in I} f_j(t)$ for all t. Then the procedure to define the set of boundary jobs M and the diagonal Δ is exactly the same as that for the linearly ordered case, except that $f^i = \text{pmax}\{f_j \,|r_j = r^i\}$ need not equal f_j for any real job j with $r_j = r^i$, rather f^i is the pointwise maximum for all t of the functions f_j with $r_j = r^i$ for $i = 1, 2, \ldots, K$.

Algorithm Δ for $1/r/f_{\max}$
Let $(r_{M_1}, f_{M_1}) = (r^1, f^1)$, $l = 1$
For $i = 2$ to K
  If $f_{M_l} \geq_f f^i$ go to Next i
  Else
    $(r_{M_{l+1}}, f_{M_{l+1}}) = (r^i, \text{pmax}\{f^i, f_{M_l}\})$ /* This defines M_{l+1} and the function $f_{M_{l+1}}$
    $(r_{D_l}, f_{D_l}) = (r^i, f_{M_l})$ /* This defines D_l and the function f_{D_l}
    Increase l to $l+1$
Next i

We can represent the adjacent interchange order $\lessdot$ using a 3-dimensional structure. Here jobs are represented by curves in space resting on planes determined by their release times, where $k \lessdot m$ ($\iff r_k \leq r_m$ and $f_k \geq f_m$) if k is on a lower r-plane than m and f_k is above f_m, i.e. $f_k(t) \geq f_m(t)$ *for all* t. Similarly, Δ and M define a staircase-like structure that contains all of the curves f_j for $j \in J$ inside or on its surface (see Figure 8 for an example with $|\Delta| = 4$). The definition of $\prec$ is analogous to the previous special cases, i.e. $k \prec m \iff k$ and m are Δ-separated.

Theorem 6 *$\prec$ is a precedence order for* $1/r/f_{\max}$.

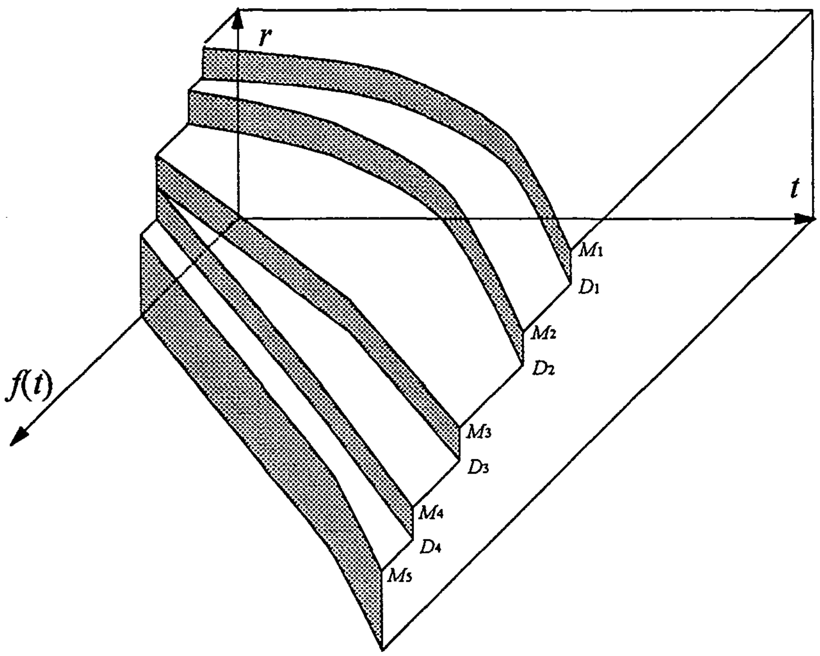

Figure 8: Staircase structure for the general case.

Proof. The proof is the same as that for $1\,/r\,/W L_{\max}$. ■

Finally, we note that even though the $\prec$ precedence order of Theorem 6 applies to very general $f_{\max}$ problems, its structure, from an order-theoretic point of view, is *not* different from the case when $\geq_f$ is linearly ordered: Let us further augment P_Δ by adding new least and greatest elements 0 and 1, and call it $P_{0,1} = (J_{0,1}, <_{0,1})$, where $J_{0,1} = J_\Delta \cup \{0,1\}$ and $<_{0,1} = <_\Delta \cup (\{0\} \times J_\Delta) \cup (J_\Delta \times \{1\})$. Then $\prec$ admits an interval representation using intervals $I_j = [x(j), y(j)]$ $(j \in J)$ on the set $S = \Delta \cup \{0,1\}$ linearly ordered by $<_{0,1}$, where for $j \in J$, $x(j) = \max\{l \in S\,|l< j\}$ and $y(j) = \min\{l \in S\,|j< l\}$. This leads to the following corollary for $1\,/r\,/f_{\max}$.

Corollary 3 $1\,/r, prec\,/f_{\max}$ *remains NP-hard in the strong sense even with interval order precedence constraints.*

References

[1] Dushnik, B., and E.W. Miller (1941). Partially ordered sets. *Amer. J. Math.* 63, 600-610.

[2] Emmons, H. (1969). One machine sequencing to minimize certain functions of job tardiness. *Oper. Res.* 17, 701-715.

[3] Erschler, J., G. Fontan, C. Merce and F. Roubellat (1982). Applying new dominance concepts to job schedule optimization. *European J. Oper. Res.* 11, 60-66.

[4] Erschler, J., G. Fontan, C. Merce and F. Roubellat (1983). A new dominance concept in scheduling n on a single machine with ready times and due dates. *Oper. Res.* 31, 114-127.

[5] Fishburn, P.C. (1985) *Interval Orders and Interval Graphs*, J. Wiley, New York, N.Y.

[6] Fontan, G. (1980). Notion de dominance et son application à l'étude de certains problèmes d'ordonnancement, Thèse de Doctorat ès Sciences, Université Paul Sabatier, Toulouse, France.

[7] Hall, L.A. (1993). A note on generalizing the maximum lateness criterion for scheduling. *Discrete Appl. Math.* 47, 129-137.

[8] Jackson, J.R. (1955). Scheduling a production line to minimize maximum tardiness, Research Report 43, Management Science Research Project, Univ. of California, Los Angeles.

[9] Johnson, S.M. (1954). Optimal two- and three-stage production schedules with setup times included. *Naval Res. Logist. Quart.* 1, 61-68.

[10] Lageweg, B.J., J.K. Lenstra and A.H.G. Rinnooy Kan (1976). Minimizing maximum lateness on one machine: computuational experience and some applications. *Stast. Neerlandica* 30, 25-41.

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

[12] Lawler, E.L., J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys (1989). Sequencing and scheduling: Algorithms and complexity. Report BS-R8909, CWI Amsterdam.

[13] Merce, C. (1979). Etude de l'existence de solutions pour certains problèmes d'ordonnancement, Thèse de Doctorat Ingénieur, Université Paul Sabatier, Toulouse, France.

[14] Monma, C.L. (1978). Properties and efficient algorithms for certain classes of sequencing problems, Ph. D. Thesis, Cornell University, Ithaca, New York.

[15] Papadimitriu, C.H. and M. Yannakakis (1979). Scheduling interval-ordered tasks. *SIAM J. Computing* 8, 405-409.

[16] Pinedo, M. (1995) *SCHEDULING: Theory, Algorithms and Systems*, Prentice Hall, Englewood Cliffs, N.J.

[17] Potts, C.N. (1980). Analysis of a heuristic for one machine sequencing with release dates and delivery times. *Oper. Res.* 28, 1436-1441.

[18] Smith, W.E. (1956). Various optimizers for single-stage production. *Naval Res. Logist. Quart.* 3, 59-66.