MATHEMATICS OF OPERATIONS RESEARCH
Vol. 30, No. 4, November 2005, pp. 1005–1021
ISSN 0364-765X | EISSN 1526-5471 | 05 | 3004 | 1005

DOI 10.1287/moor.1050.0158

Single-Machine Scheduling with Precedence Constraints

José R. Correa
School of Business, Universidad Adolfo Ibáñez, Avenida Presidente Errázuriz 3485,
Las Condes, Santiago, Chile, correa@uai.cl, http://jcorrea.uai.cl/

Andreas S. Schulz
Sloan School of Management, Massachusetts Institute of Technology, Office E53-361,
77 Massachusetts Avenue, Cambridge, Massachusetts 02139-4307, schulz@mit.edu, http://web.mit.edu/schulz/

We discuss the problem of sequencing precedence-constrained jobs on a single machine to minimize the average weighted completion time. This problem has attracted much attention in the mathematical programming community since Sidney's pioneering work in 1975 (Sidney, J. B. 1975. Decomposition algorithms for single machine scheduling with precedence relations and deferral costs. *Operations Research* **23** 283–298). We look at the problem from a polyhedral perspective and uncover a relation between Sidney's decomposition theorem and different linear programming relaxations. More specifically, we present a generalization of Sidney's result, which particularly allows us to reason that virtually all known 2-approximation algorithms are consistent with his decomposition. Moreover, we establish a connection between the single-machine scheduling problem and the vertex cover problem. Indeed, in the special case of series-parallel precedence constraints, we prove that the sequencing problem can be seen as a special case of the vertex cover problem. We also argue that this result is true for general precedence constraints if one can show that a certain integer program represents a valid formulation of the sequencing problem. Finally, we give a 3/2-approximation algorithm for two-dimensional partial orders, and we also provide a characterization of the active inequalities of a linear programming relaxation in completion time variables.

Key words: linear ordering; scheduling; vertex cover; linear programming relaxation; integer programming formulation; approximation algorithm; partial order
MSC2000 subject classification: Primary: 90B35, 90C57, 90C59; secondary: 06A07, 68W25, 90C08, 90C09, 90C27
OR/MS subject classification: Primary: Production/scheduling: sequencing, deterministic, single machine; secondary: mathematics, combinatorics
History: Received August 18, 2004; revised December 27, 2004.

1. Introduction. We consider the following scheduling problem. A set $N = \{1, \dots, n\}$ of n jobs has to be processed on a single machine, which can handle at most one job at a time. Each job j has a positive processing time, $p_j > 0$, and a nonnegative weight, $w_j \geqslant 0$, and we want to find a schedule of the jobs that minimizes the weighted sum of job completion times, $\sum_{j\in N} w_j C_j$. Here, C_j denotes the time at which job j is completed in a feasible schedule. In this basic form, the problem can be solved efficiently using Smith's rule [33], which sequences the jobs in nonincreasing order of their ratios w_j/p_j of weight to processing time. In this paper, we focus on the case when the jobs have to be consistent with precedence constraints. The precedence constraints are given in the form of a directed acyclic graph (i.e., a partial order) $G = (N, P)$, where $(i, j) \in P$ implies that job i must be completed before job j can be started. We assume that G is transitively closed; i.e., if $(i, j), (j, k) \in P$, then $(i, k) \in P$. In standard scheduling notation (Graham et al. [10]), this problem is known as $1|\text{prec}|\sum w_j C_j$. Lawler [15] and Lenstra and Rinnooy Kan [16] showed that this problem is strongly NP-hard.

Several integer programming formulations and linear programming relaxations have been proposed for this problem. They can basically be divided into three groups according to the decision variables used: some formulations exploit time-indexed variables (e.g., Dyer and Wolsey [7]), which are binary variables indicating when a job is completed; others make direct use of completion time variables (e.g., Balas [1]); and yet others borrow their decision variables from the underlying linear ordering polytope (e.g., Wolsey [36]). We refer to Queyranne and Schulz [27] for an overview and a collection of further references.

Linear programming relaxations in these variables have been successfully used to obtain constant-factor approximation algorithms for this problem.[1] The first one, proposed by

[1] An α-approximation algorithm runs in polynomial time and produces for every instance a feasible schedule of cost at most α times that of an optimal schedule. The value α is called the performance guarantee of the algorithm.

Hall et al. [11], relies on a time-indexed linear programming relaxation and has performance guarantee $4+\varepsilon$. Subsequently, Schulz [30] presented a 2-approximation algorithm based on solving a weaker linear programming relaxation in completion time variables; see also Hall et al. [12]. The analysis also implies that a linear programming relaxation in linear ordering variables suggested by Potts [25] can be used to obtain another 2-approximation algorithm. Later on, Chudak and Hochbaum [5] proposed a relaxation of Potts' linear program that suffices to get yet another approximation algorithm of the same performance guarantee. Moreover, they showed that the weaker linear program can be solved by one min-cut computation, which yields a combinatorial 2-approximation algorithm. Independently, Chekuri and Motwani [4] and Margot et al. [17] used Sidney's decomposition theorem [32] to give an entire family of combinatorial 2-approximation algorithms. Afterwards, Goemans and Williamson [9] revived the two-dimensional Gantt charts of Eastman et al. [8] to illustrate the findings of Margot et al. [17] and Chekuri and Motwani [4]. They also proved the correctness of Lawler's polynomial-time, exact algorithm for series-parallel precedence constraints [15] by relating it to the dual of a linear programming relaxation in completion time variables due to Queyranne and Wang [28]. Woeginger [34] argued that the approximability behavior of $1|\text{prec}|\sum w_jC_j$ and that of several of its (NP-hard) special cases (e.g., precedence constraints of height 1 where all minimal jobs have unit processing time and zero weight while all maximal jobs have zero processing time and unit weight) is essentially identical. He also explored a relationship between $1|\text{prec}|\sum w_jC_j$ and the partially-ordered knapsack problem, which can be used to derive $(1.618+\varepsilon)$-approximation algorithms for particular classes of precedence constraints, including interval orders and two-dimensional orders. The latter result is due to Kolliopoulos and Steiner [14].

After setting the stage in §2, we establish in §3.1 a connection between Sidney's decomposition theory and the linear programming relaxations in linear ordering variables by Potts [25] and Chudak and Hochbaum [5]. In §3.2, we propose a new relaxation in linear ordering variables that suggests a strong relationship between $1|\text{prec}|\sum w_jC_j$ and the vertex cover problem. We also show that the sequencing problem is indeed a special case of the vertex cover problem when the precedence constraints are series-parallel (§3.3). In §3.4, we show that the new linear ordering relaxation is integer if and only if the precedence constraints are two-dimensional. We also give a simple 3/2-approximation algorithm for the class of two-dimensional precedence constraints. In §3.5, we provide a bound on the optimal value of the considered linear ordering relaxations, which implies that the family of 2-approximation algorithms by Chekuri and Motwani [4] and Margot et al. [17] already has performance guarantee 2 for the weighted sum of starting times objective. We study relaxations in completion time variables in §4, extending results by Margot et al. [17] on the structure of optimal solutions. The results in §§3.1 and 4 imply that all known 2-approximation algorithms follow Sidney's decomposition and are therefore special cases of the class of algorithms described by Chekuri and Motwani [4] and Margot et al. [17].

2. Definitions and preliminaries. For a job $j\in N$, we denote the ratio w_j/p_j by ρ_j. We generalize these quantities to sets $S\subseteq N$ of jobs in the usual way: $p(S):=\sum_{j\in S}p_j$, $w(S):=\sum_{j\in S}w_j$, and $\rho(S):=w(S)/p(S)$. For $S\subseteq N$, G_S denotes the subgraph of G induced by S, and $P(G_S)$ is the set of arcs (precedence constraints) in G_S. A set of jobs $I\subseteq S$ is called *initial* in G_S if $j\in I$ and $(i,j)\in P(G_S)$ imply $i\in I$. Analogously, $F\subseteq S$ is called *final* in G_S if $S\backslash F$ is initial in G_S. We simply say that $S\subseteq N$ is initial (respectively, final) if S is initial (final) in G. If there exists some final set $F\subseteq N$ such that $w(F)=0$, the jobs in F can be scheduled in an arbitrary feasible sequence after all jobs in $N\backslash F$ without affecting the objective function value. We assume for the rest of this paper that $w(F)>0$ for all final sets $F\subseteq N$.

A nonempty set $S^*\subseteq S$ is a ρ-*maximal* initial set in G_S if S^* is an initial set in G_S with maximum value of ρ. In other words, $S^*\in\arg\max\{\rho(S')\colon\ S'\neq\varnothing \text{ initial in } G_S\}$. An initial set $S \subseteq N$ is said to be *non-Sidney-decomposable* if the only ρ-maximal initial set contained in S is S itself; i.e., S is a minimal (with respect to inclusion) ρ-maximal initial set.

LEMMA 2.1 (SIDNEY [32]). *If $S \subseteq N$ is a ρ-maximal initial set, F is final in G_S, and I is initial in $G_{N\setminus S}$, then $\rho(I) \leqslant \rho(S) \leqslant \rho(F)$.*

PROOF. It suffices to note that for two disjoint sets A and B of jobs, $\rho(A \cup B)$ can be written as a convex combination of $\rho(A)$ and $\rho(B)$. Indeed,

$$\rho(A \cup B) = \frac{w(A)+w(B)}{p(A)+p(B)} = \rho(A)\frac{p(A)}{p(A\cup B)} + \rho(B)\frac{p(B)}{p(A\cup B)}. \quad \square$$

We now review the concept of Sidney decomposition. Consider a partition of N into disjoint sets $S_1, S_2, \ldots, S_k$ such that each S_i is a ρ-maximal initial set of $G_{S_i\cup\cdots\cup S_k} = G_{N\setminus\{S_1\cup\cdots\cup S_{i-1}\}}$, for $i = 1, \ldots, k$. The partition $(S_1, S_2, \ldots, S_k)$ is called a *Sidney decomposition* of N. Sidney [32] proved that there exists an optimal solution to $1|\text{prec}|\sum w_j C_j$ that processes the jobs in S_i before those in S_j, whenever $i < j$. This result is known as *Sidney's decomposition theorem*. A Sidney decomposition is in general not unique. However, Lemma 2.1 implies that given a Sidney decomposition $(S_1, S_2, \ldots, S_k)$, if $\rho(S_i) > \rho(S_j)$, then $i < j$. Margot et al. [17] introduced the ρ-profile of a Sidney decomposition $(S_1, S_2, \ldots, S_k)$ as the decreasing sequence $\lambda_1 > \cdots > \lambda_q$ of distinct values $\rho(S_i)$ and showed that all Sidney decompositions of a given instance have the same ρ-profile. In particular, there is a unique coarsest Sidney decomposition, which they called the *reduced Sidney decomposition*: $(R_1, R_2, \ldots, R_q)$ with $R_i := \bigcup\{S_j\colon \rho(S_j) = \lambda_i\}$ for $i = 1, \ldots, q$. We say that a scheduling algorithm *is consistent* with Sidney's decomposition if, for the reduced Sidney decomposition $(R_1, R_2, \ldots, R_q)$, the algorithm schedules the jobs in R_i before the ones in R_j, whenever $i < j$. The reduced Sidney decomposition can be computed in polynomial time (Lawler [15], Picard and Queyranne [23], and Margot et al. [17]).

Let us now introduce the linear programming relaxations of $1|\text{prec}|\sum w_i C_j$, which will be analyzed in subsequent sections. The first, due to Potts [25], is based on an integer programming formulation using linear ordering variables δ_{ij}. The variable δ_{ij} has value 1 if job i precedes job j in the corresponding schedule, and 0 otherwise.

$$[\text{P}] \qquad \min \quad \sum_{j\in N} p_j w_j + \sum_{i,\,j\in N} p_i w_j \delta_{ij} \tag{1a}$$

$$\text{s.t.} \quad \delta_{ij} + \delta_{ji} = 1 \quad \text{for all } i, j \in N, \tag{1b}$$

$$\delta_{ij} + \delta_{jk} + \delta_{ki} \geqslant 1 \quad \text{for all } i, j, k \in N, \tag{1c}$$

$$\delta_{ij} = 1 \quad \text{for all } (i, j) \in P, \tag{1d}$$

$$\delta_{ij} \in \{0, 1\} \quad \text{for all } i, j \in N. \tag{1e}$$

To simplify notation, we implicitly assume that this and future formulations do not contain variables of the form δ_{jj} for $j \in N$. It is easy to see that any feasible solution δ to the above integer program represents a valid schedule and that the objective function value of δ coincides with the total weighted completion time of that schedule. Chudak and Hochbaum [5] proposed to study the following relaxation of [P]:

$$[\text{CH}] \qquad \min\,(1\text{a})$$

$$\text{subject to } (1\text{b}),\ (1\text{d}),\ (1\text{e}), \quad \text{and}$$

$$\delta_{ik} + \delta_{kj} \geqslant 1 \quad \text{for all } (i, j) \in P,\ k \in N. \tag{2}$$

In other words, [CH] just keeps those transitivity constraints (1c) for which two of the participating jobs are already related to each other by a precedence constraint.

This new integer program leads to two natural questions, already raised by Chudak and Hochbaum [5]: Is an optimal solution of [P] also optimal for [CH]? In other words, is [CH] a valid formulation of the scheduling problem (in the sense that it gives the right objective function value and there exists an optimal solution that is a schedule)? Moreover, if [P-LP] and [CH-LP] are the linear programming relaxations of [P] and [CH], respectively, is an optimal solution to [P-LP] optimal for [CH-LP]? Let us formulate these questions as conjectures:

CONJECTURE 2.1. *An optimal solution to* [P] *is optimal for* [CH] *as well.*

CONJECTURE 2.2. *An optimal solution to* [P-LP] *is optimal for* [CH-LP] *as well.*

Both conjectures are true if the set of precedence constraints is empty (Wolsey [36]) or series-parallel (see Theorem 3.2 below). If true in general, they would lead to several interesting consequences, as we will point out in the remainder of this paper. Moreover, we will also prove a number of results providing evidence in support of these conjectures.

In §3.2 we will prove that [CH] is a special case of the vertex cover problem. A *vertex cover* of an undirected graph $H = (V, E)$ is a subset $C \subseteq V$ of nodes that contains at least one endpoint of every edge. The *vertex cover problem* is that of finding a vertex cover C of minimum total weight $w(C) = \sum_{v \in C} w_v$ in a graph with nonnegative node weights $w_v \geqslant 0$. The classic integer program to formulate the vertex cover problem is

$$\begin{aligned} \min \quad & \sum_{v \in V} w_v x_v \\ \text{s.t.} \quad & x_u + x_v \geqslant 1 \quad \text{for all } \{u, v\} \in E, \\ & x_v \in \{0, 1\} \quad \text{for all } v \in V. \end{aligned}$$

If we relax the integrality constraints and replace them by $x_v \geqslant 0$ for all $v \in V$, then we obtain a linear programming relaxation that is usually referred to as the vertex cover LP. Nemhauser and Trotter [19, 20] proved that this relaxation is half-integral (i.e., all basic feasible solutions have coordinates which are either 0, 1/2, or 1) and that an optimal solution can be obtained via a single min-cut computation. Moreover, if x is an optimal solution to the vertex cover LP, then there exists an optimal vertex cover that contains v for all nodes $v \in V$ with $x_v = 1$, and it does not contain v for all nodes $v \in V$ with $x_v = 0$. This is known as the *persistency property* of the vertex cover problem.

Let us also introduce a linear programming relaxation in completion time variables, which uses the following additional notation: $p^2(S) := \sum_{j \in S} p_j^2$ for $S \subseteq N$.

[QW-LP]

$$\min \quad \sum_{j \in N} w_j C_j \tag{3a}$$

$$\text{s.t.} \quad \sum_{j \in S} p_j C_j \geqslant \frac{1}{2}\big(p(S)^2 + p^2(S)\big) \quad \text{for all } S \subseteq N, \tag{3b}$$

$$C_j - C_i \geqslant p_j \quad \text{for all } (i, j) \in P. \tag{3c}$$

Inequalities (3b) are known as the *parallel inequalities*; they suffice to describe the convex hull of feasible completion time vectors in the absence of precedence constraints (Wolsey [35] and Queyranne [26]). Inequalities (3c) model the precedence constraints.

Finally, we briefly describe the known classes of approximation algorithms for $1|\text{prec}|\sum w_j C_j$ with a performance guarantee of 2:

(A) Let C be an optimal solution to [QW-LP]. Schedule the jobs in nondecreasing order of C_j, breaking ties arbitrarily (Schulz [30]).

(B) Let δ be an optimal solution to [CH-LP]. Compute $C_j := \sum_{i \in N} p_i \delta_{ij} + p_j$ and schedule the jobs in nondecreasing order of C_j, breaking ties arbitrarily (Chudak and Hochbaum [5]).[2]

[2] In a precursor to this algorithm, δ was chosen to be an optimal solution to [P-LP]; see Schulz [30] and Hall et al. [12] for details.

(C) Compute a Sidney decomposition of $G=(N,P)$. Schedule the jobs in any feasible order consistent with this decomposition (Chekuri and Motwani [4] and Margot et al. [17]).

It is worth mentioning that time-indexed linear programming relaxations have also been used to find approximate solutions for this problem (Hall et al. [11], Schulz [30], Hall et al. [12], and Schulz and Skutella [31]). However, these algorithms are either nonpolynomial or have a performance guarantee (slightly) worse than 2.

3. Linear ordering relaxations. In this section, we consider a variety of formulations and linear programming relaxations of $1|\text{prec}|\sum w_j C_j$ in linear ordering variables. In §3.1, we prove a structural characteristic of the optimal solutions of [CH-LP], [P-LP], [CH], and [P], which generalizes Sidney's decomposition theorem. We propose a new linear programming relaxation in §3.2; while it is equivalent to [CH-LP], it helps to uncover the connection to the vertex cover problem. We study special classes of precedence constraints in §§3.3 and 3.4, and derive a bound on the optimal value of linear ordering relaxations in §3.5.

3.1. A structural result.

THEOREM 3.1. *Let $S\subset N$ be a ρ-maximal initial set of jobs. Then, each of the following mathematical programming formulations has an optimal solution δ such that $\delta_{ij}=1$ for all $i\in S$, $j\in N\setminus S$:* [CH-LP], [CH], [P-LP], *and* [P]. *Moreover, if no superset T of S satisfies $\rho(T)=\rho(S)$, then every optimal solution δ satisfies $\delta_{ij}=1$ for all $i\in S$, $j\in N\backslash S$.*

PROOF. Let δ be an optimal solution of the considered mathematical program [X]. Suppose that $\delta_{ij}<1$ for some $i\in S, j\in N\backslash S$. For each $k\in S$, define the set $I_k:=\{j\in N\backslash S\colon \delta_{jk}>0\}$. Similarly, for each $k\in N\backslash S$, let $F_k:=\{i\in S\colon \delta_{ki}>0\}$. Because δ satisfies (2), each F_k is a final set in the graph G_S and each I_k is an initial set in $G_{N\backslash S}$. Let $\varepsilon:=\min\{\delta_{ij}\colon i\in N\backslash S,\ j\in S,\ \delta_{ij}>0\}$ and consider the vector δ' defined as

$$\delta'_{ij}:=\begin{cases}\delta_{ij}+\varepsilon & \text{if } i\in S,\ j\in N\backslash S \text{ and } \delta_{ij}<1,\\ \delta_{ij}-\varepsilon & \text{if } i\in N\backslash S,\ j\in S \text{ and } \delta_{ij}>0,\\ \delta_{ij} & \text{otherwise,}\end{cases}\qquad \text{for } i,j\in N.$$

Clearly, $0\leqslant\delta'_{ij}\leqslant 1$, and δ' satisfies (1b) and (1d). Moreover, if [X] is an integer program, then $\varepsilon=1$ and δ' is integer, too. We argue next that δ' also satisfies (2) (or even (1c) in case of [P] and [P-LP]). Let $(i,j)\in P$ and $k\in N$. If either $i,j\in S$ or $i,j\in N\backslash S$, this holds trivially. If $i\in S$ and $j\in N\backslash S$, then either δ_{ik} or δ_{kj} was incremented by ε or both are unchanged; hence, $\delta'_{ik}+\delta'_{kj}\geqslant\delta_{ik}+\delta_{kj}\geqslant 1$. The case $j\in S$, $i\in N\backslash S$ does not occur because S is initial. It follows that δ' is feasible for [CH-LP] or [CH]. If there is no precedence relation between any two of three jobs $i,j,k\in N$, the triangle constraints (1c) are satisfied by δ' if they are satisfied by δ. Indeed, each 3-cycle (i,j,k), which is neither completely contained in S nor in $N\backslash S$, has exactly one forward arc and one backward arc across the cut $(S,N\backslash S)$. Hence, δ' is feasible for [X].

The difference in the objective function values of δ and δ' can be calculated as follows:

$$\begin{aligned}\sum_{i,j\in N}p_iw_j\delta_{ij}-\sum_{i,j\in N}p_iw_j\delta'_{ij}&=\varepsilon\sum_{k\notin S}p_kw(F_k)-\varepsilon\sum_{k\in S}p_kw(I_k)\\&=\varepsilon\sum_{k\notin S}p_kp(F_k)\rho(F_k)-\varepsilon\sum_{k\in S}p_kp(I_k)\rho(I_k).\end{aligned}$$

By applying Lemma 2.1 to the sets I_k and F_k, we obtain $\rho(I_k)\leqslant\rho(S)\leqslant\rho(F_k)$. Hence, the above quantity can be bounded from below by

$$\varepsilon\rho(S)\left(\sum_{k\notin S}p_kp(F_k)-\sum_{k'\in S}p_{k'}p(I_{k'})\right).$$

Because $k' \in F_k$ if and only if $k \in I_{k'}$, this expression evaluates to zero. Therefore, the objective function value of δ' is not worse than that of δ. Moreover, the variable that determined the value of ε has been reduced to 0. After at most $O(n^2)$ iterations of this procedure, we obtain the solution we were looking for.

For the additional remark, note that $\rho(I_k) < \rho(S)$ if no superset of S is ρ-maximal. Hence, if an optimal solution δ satisfies $\delta_{ij} < 1$ for some $i \in S$, $j \in N \setminus S$, then the procedure above would result in a solution of strictly smaller objective function value, a contradiction. $\square$

Because [P] is an actual formulation of the scheduling problem $1|\text{prec}|\sum w_j C_j$, we obtain Sidney's decomposition theorem as a corollary to Theorem 3.1.

COROLLARY 3.1 (SIDNEY [32]). *Let $(S_1, S_2, \ldots, S_k)$ be a Sidney decomposition of $G = (N, P)$. Then, there exists an optimal sequence for N in which the jobs in S_i are a consecutive subsequence, succeeding all jobs in $S_1 \cup \cdots \cup S_{i-1}$ and preceding all jobs in $S_{i+1} \cup \cdots \cup S_k$, for $i = 1, 2, \ldots, k$.*

PROOF. Applying Theorem 3.1 iteratively to the sets $N, N\setminus S_1, N\setminus(S_1 \cup S_2), \ldots$ implies that there exists an optimal solution δ to [P] such that $\delta_{ij} = 1$ for all $i \in S_k$, $j \in S_\ell$ with $k < \ell$. $\square$

Moreover, each optimal sequence has to be consistent with the reduced Sidney decomposition. According to Theorem 3.1, Sidney's decomposition is already a feature of the linear programming relaxations [P-LP] and [CH-LP]. Consequently, Algorithm (B) belongs in fact to the family (C) of algorithms.

COROLLARY 3.2. *Algorithm* (B) *is consistent with Sidney's decomposition.*[3]

Let us conclude this section by pointing out that the statement of Theorem 3.1 for the integer program [CH] had to be true if Conjecture 2.1 is true. Indeed, if [CH] is a formulation of the scheduling problem $1|\text{prec}|\sum w_j C_j$, then the corresponding part of Theorem 3.1 is implied by Sidney's decomposition theorem.

3.2. A new linear programming relaxation. We now propose a new linear ordering relaxation of $1|\text{prec}|\sum w_j C_j$, which can be interpreted as a vertex cover problem. We also prove that this formulation is actually equivalent to [CH]. The integer program is as follows:

$$
\begin{aligned}
&[\text{CS}] \qquad && \min\,(1a)\\
& && \text{subject to (1d), (1e), (2)}, \ \delta_{ij} = 0 \quad \text{for } (j, i) \in P, \ \text{and}\\
& && \quad \delta_{ij} + \delta_{ji} \geqslant 1 \quad \text{for all } i, j \in N,\\
& && \quad \delta_{i\ell} + \delta_{kj} \geqslant 1 \quad \text{for all } (i, j), (k, \ell) \in P.
\end{aligned}
$$

As usual, let us denote by [CS-LP] the linear relaxation of this integer program. Because we can obviously omit the variables δ_{ij} and δ_{ji} for $(i, j) \in P$ from the formulation, [CS-LP] is equivalent to

$$
[\text{CS}'\text{-LP}] \qquad \min \quad \sum_{\substack{i,j \in N\\ i \| j}} p_i w_j \delta_{ij} + \sum_{j \in N} p_j w_j + \sum_{(i,j) \in P} p_i w_j \tag{4a}
$$

$$
\text{s.t.} \quad \delta_{ij} + \delta_{ji} \geqslant 1 \quad \text{for all } i, j \in N, \ i \| j, \tag{4b}
$$

$$
\delta_{ik} + \delta_{kj} \geqslant 1 \quad \text{for all } (i, j) \in P, \ k \in N, \ i \| k \| j, \tag{4c}
$$

$$
\delta_{i\ell} + \delta_{kj} \geqslant 1 \quad \text{for all } (i, j), (k, \ell) \in P, \ i \| \ell \ \text{ and } \ j \| k, \tag{4d}
$$

$$
\delta_{ij} \geqslant 0 \quad \text{for all } i, j \in N, \ i \| j. \tag{4e}
$$

[3] Technically, Chudak and Hochbaum [5] proposed a second algorithm, which is based on ordering the jobs according to their fractional completion times in the instance where processing times and weights are swapped and precedence constraints reversed. It follows from Theorem 3.1 that this algorithm is consistent with Sidney's decomposition as well.

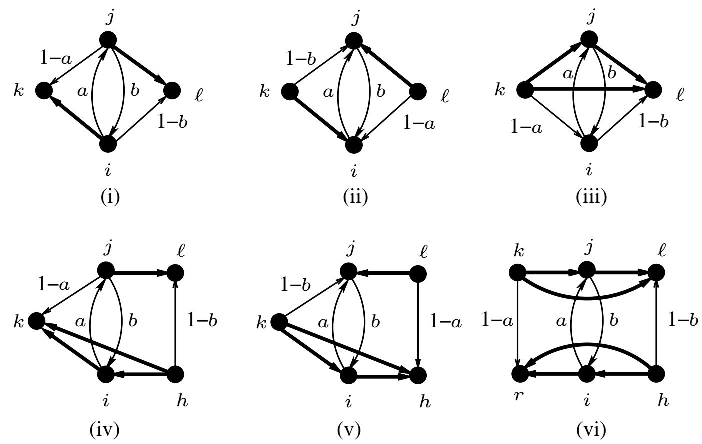

FIGURE 1. Relevant cases (i)–(vi) in the proof of Lemma 3.1.

Here, $i \parallel j$ means that neither (i, j) nor (j, i) belongs to P. [CS′] can be interpreted as a vertex cover problem in an undirected graph $G_{CS}(P)$ that has a node for each ordered pair (i, j) of jobs $i, j \in N$ with $i \parallel j$. Two nodes (i, j) and (k, ℓ) are adjacent if either $j = k$ and $i = \ell$, or $j = k$ and $(i, \ell) \in P$, or $(i, \ell), (k, j) \in P$. Let us show next that [CS′-LP] is the dominant[4] of [CH′-LP], where [CH′-LP] is the following linear program equivalent to [CH-LP]:

$$\text{[CH}'\text{-LP]} \qquad \min (4a) \quad \text{subject to } (4c), (4e), \text{ and } \delta_{ij} + \delta_{ji} = 1 \quad \text{for all } i, j \in N,\ i \parallel j.$$

The following result makes no use of the special structure of the coefficients in (4a); it holds for all positive objective functions in the δ_{ij} variables.

LEMMA 3.1. *The optimal solutions to* [CH-LP] *and the optimal solutions to* [CS-LP] *coincide. Moreover,* [CS′-LP] *is the dominant of* [CH′-LP].

PROOF. Let δ be a feasible solution to [CH′-LP] and assume that $(i, j), (k, \ell) \in P$, $i \parallel \ell$, $j \parallel k$. Because δ satisfies (4c), it follows that $2\delta_{kj} + 2\delta_{i\ell} + \delta_{\ell j} + \delta_{j\ell} + \delta_{ki} + \delta_{ik} \geqslant 4$. As δ also satisfies $\delta_{\ell j} + \delta_{j\ell} = 1$ and $\delta_{ki} + \delta_{ik} = 1$, we can infer that δ satisfies (4d).

On the other hand, let δ be a feasible solution to [CS′-LP] such that $\delta_{ij} + \delta_{ji} > 1$ for some $i, j \in N, i \parallel j$. Say $\delta_{ij} = a$ and $\delta_{ji} = b$, with $a + b > 1$. We claim that either δ_{ij} or δ_{ji} (or both) can be reduced without destroying feasibility. Suppose not. Then, (i, j) and (j, i) must each belong to a tight inequality of the form (4c) or (4d). This leads to six basic cases, which are depicted in Figure 1. Here, bold-faced arcs represent precedence constraints, and the remaining arcs correspond to variables. All other possible cases arise from exchanging the roles of i and j and can therefore be handled analogously.

In cases (i)–(iii), both (i, j) and (j, i) belong to a tight inequality of the form (4c); cases (iv) and (v) refer to the situation in which (i, j) (or (j, i)) is in a tight inequality (4c), while (j, i) (or (i, j)) is part of a tight inequality (4d); if (i, j) and (j, i) each belong to a tight inequality (4d), then we are in case (vi). We can find a violated inequality in each case: (i) $\delta_{jk} + \delta_{i\ell} = 2 - (a + b) < 1$; (ii) $\delta_{kj} + \delta_{\ell i} = 2 - (a + b) < 1$; (iii) $\delta_{ki} + \delta_{i\ell} = 2 - (a + b) < 1$; (iv) $\delta_{jk} + \delta_{h\ell} = 2 - (a + b) < 1$; (v) $\delta_{kj} + \delta_{\ell h} = 2 - (a + b) < 1$; and (vi) $\delta_{kr} + \delta_{h\ell} = 2 - (a + b) < 1$. As this contradicts the feasibility of δ, it follows that the value of δ_{ij} or δ_{ji} can be decreased until $\delta_{ij} + \delta_{ji} = 1$. As a result, [CS′-LP] is the dominant of [CH′-LP]. In particular, an optimal solution to [CS-LP] satisfies (1b) and hence is feasible (and optimal) for [CH-LP]. □

[4] A linear program is the dominant of another linear program if its feasible region is the Minkowski sum of the feasible region of the other linear program and the nonnegative orthant.

The previous lemma implies that Theorem 3.1 is also valid for [CS-LP] and [CS]. Moreover, as [CS] represents an instance of the vertex cover problem, it follows from the work of Nemhauser and Trotter [19, 20] that [CS-LP] is half-integral and that an optimal LP solution can be obtained via a single min-cut computation. Hence, the same holds for [CH-LP], which implies Theorem 2.4 in Chudak and Hochbaum's paper [5].

Interestingly, Theorem 3.1 also implies that we can (possibly) get a refinement of the reduced Sidney decomposition as follows:

(a) Let δ be an optimal solution to [CS-LP].
(b) Consider the digraph $D = (N, A)$ with arc set $A := \{(i, j)\colon \delta_{ij} = 1\}$.
(c) Remove from D all arcs that belong to a cycle.
(d) Compute the series decomposition $(S_1, S_2, \ldots, S_k)$ of the remaining digraph.[5]

It follows from Theorem 3.1 that $(S_1, S_2, \ldots, S_k)$ is a refinement of the reduced Sidney decomposition. In particular, every feasible job sequence that is consistent with this order is a 2-approximation.

In this light, it is of course tempting to conjecture that what the Sidney decomposition does for the scheduling problem is indeed equivalent to what the persistency property (Nemhauser and Trotter [20]) brings about for the vertex cover problem. In particular, we know that every feasible schedule that is consistent with a Sidney decomposition is a 2-approximation for the scheduling problem (Chekuri and Motwani [4] and Margot et al. [17]), while every feasible vertex cover that includes all variables that are equal to one and that does not contain any variable that is equal to zero in an optimal basic feasible solution to the vertex cover LP, is a 2-approximation for the vertex cover problem (Hochbaum [13]). In fact, one might conjecture that the following is true:

> Let δ be a unique optimal solution to [CH-LP]. Then, there exists an optimal schedule in which job i precedes job j whenever $\delta_{ij} = 1$.

Here, the uniqueness assumption is necessary because, even if Conjecture 2.1 is true, optimal solutions to [CH] can in general contain cycles and thus do not represent valid schedules. (For example, take three identical jobs without any precedence relations.) However, short of a proof of Conjecture 2.1, we have to confine ourselves to the following result.

COROLLARY 3.3. *If δ is an optimal solution to* [CS-LP], *then there exists an optimal solution δ' to its integer counterpart* [CS] *such that $\delta'_{ij} = 1$ whenever $\delta_{ij} = 1$, and $\delta'_{ij} = 0$ whenever $\delta_{ij} = 0$. Moreover, $\delta'_{ij} + \delta'_{ji} = 1$ for all $i, j \in N$.*

PROOF. The corollary is a direct consequence of the persistency property of the vertex cover problem and the proof of Lemma 3.1. □

Of course, if Conjecture 2.1 is true, the proof of Lemma 3.1 also implies that $1|\text{prec}|\sum w_j C_j$ is a special case of the vertex cover problem.

Let us finally point out that a similar analysis to that in the proof of Lemma 3.1 shows that the dominant of [P-LP] is the following linear program:

$$[\text{P}'\text{-LP}] \qquad \min \sum_{\substack{i,\, j \in N \\ i \| j}} p_i w_j \delta_{ij} + \sum_{j \in N} p_j w_j + \sum_{(i,\, j) \in P} p_i w_j$$

$$\text{s.t.} \quad \delta(\mathscr{C}) \geqslant 1 \quad \text{for all delta-cycles } \mathscr{C},$$

$$\delta_{ij} \geqslant 0 \quad \text{for all } i, j \in N, \; i \parallel j.$$

[5] The *series decomposition* of an acyclic digraph $D = (V, A)$ is a partition $(S_1, S_2, \ldots, S_k)$ of its node set N such that $(i, j) \in A$ for all $i \in S_\ell$, $j \in S_{\ell+1}$, $\ell = 1, 2, \ldots, k-1$, and the series decomposition of the subgraph induced by each set S_ℓ is S_ℓ itself.

Here, a delta-cycle $\mathscr{C} \subseteq (N \times N)\setminus P$ is a collection of arcs such that there exists a set $P_{\mathscr{C}}$ of reversed precedence constraints (i.e., a subset of $P^{-1} := \{(i, j) \in N \times N\colon\ (j, i) \in P\}$) satisfying that $\mathscr{C} \cup P_{\mathscr{C}}$ is a directed cycle in $N \times N$. [CS′-LP] is the relaxation of [P′-LP] that only considers delta-cycles of size at most 2.

Let [LP$_i$] denote (the optimal value of) the above linear program restricted to delta-cycles of size no more than i. Then, [LP$_2$] = [CS′-LP] and [LP$_\infty$] = [P′-LP]; moreover,

$$[\mathrm{LP}_2] \leqslant [\mathrm{LP}_3] \leqslant \cdots \leqslant [\mathrm{LP}_k] \leqslant \cdots \leqslant [\mathrm{LP}_\infty].$$

Conjecture 2.2 states that this chain of inequalities is actually a chain of equalities.

3.3. Series-parallel precedence constraints. While $1|\mathrm{prec}|\sum w_j C_j$ is in general strongly NP-hard, some special cases can be solved efficiently. For example, Lawler [15] presented a polynomial-time algorithm for series-parallel precedence constraints. Moreover, Queyranne and Wang [28] gave a complete characterization of the convex hull of feasible completion time vectors, while Goemans and Williamson [9] proposed a primal-dual algorithm that unifies both results.

Series-parallel precedence constraints are defined inductively (Lawler [15]); the base elements are individual jobs. Given two series-parallel digraphs $G_1 = (N_1, P_1)$ and $G_2 = (N_2, P_2)$ such that $N_1 \cap N_2 = \varnothing$, the parallel composition of G_1 and G_2 results in a partial order on $N_1 \cup N_2$ that maintains P_1 and P_2, but does not introduce any additional precedence relationships. The series composition of G_1 and G_2 leads to a partial order on $N_1 \cup N_2$ that maintains P_1 and P_2; moreover, if $i \in N_1$ and $j \in N_2$, then i precedes j in the new partial order.

With Theorem 3.1 in place, the proof of the following result becomes remarkably simple.

Theorem 3.2. *When the precedence constraints are series-parallel,* [CH-LP] *has an optimal solution that is integer and a feasible schedule.*

Proof. We proceed by induction on the number of jobs. The result is trivial when $|N| = 1$ or $|N| = 2$. Assume that the result holds for all sets of jobs with series-parallel precedence constraints of cardinality at most n. Note that if $G = (N, P)$ is series-parallel, then any induced subgraph is series-parallel as well. Let us consider a set N of jobs with $|N| = n + 1$:

(i) If $G = (N, P)$ is a series composition of G_1 and G_2, then $|N_1|, |N_2| \leqslant n$ and the induction hypothesis applies to N_1 and N_2. Because the values δ_{ij} for $i \in N_1$ and $j \in N_2$ are fixed by the decomposition, the result holds for N.

(ii) Otherwise, $N = N_1 \cup N_2$, and $i \parallel j$ for all $i \in N_1$ and $j \in N_2$. In this case, it is straightforward to show that there is a non-Sidney-decomposable set S that is either fully contained in N_1 or in N_2 (as was already observed by Sidney [32]). By Theorem 3.1, there is an optimal solution satisfying $\delta_{ij} = 1$ for all $i \in S$, $j \in N\setminus S$. Hence, we obtain the result by applying the induction hypothesis to S and $N\setminus S$. □

Theorem 3.2 implies that $1|\mathrm{prec}|\sum w_j C_j$ is a special case of the vertex cover problem for series-parallel precedence constraints. In fact, one not only obtains the optimal value by solving [CH] (or, equivalently, [CH-LP]), but after a slight perturbation of job weights, the Sidney decomposition and the optimal solution to [CH] are unique, and an optimal schedule can therefore be computed by a single min-cut computation (for solving [CH-LP]). Let us formally state this as a corollary. For simplicity, we assume that all job weights and processing times are integers.

Corollary 3.4. *Consider an instance of* $1|prec|\sum w_j C_j$ *with series-parallel precedence constraints. If one defines new job weights* $\widetilde{w}_j := w_j + \varepsilon^{2j}$, *where* $\varepsilon > 0$ *is chosen such that* $\varepsilon < 1/(2p(N))$, *then the perturbed instance has a unique Sidney decomposition as well as a unique optimal solution to* [CH-LP], *which is integer, a feasible schedule, and optimal for the original instance.*

Proof. Let us show first that the Sidney decomposition of the perturbed instance is unique and a refinement of the reduced Sidney decomposition of the original instance. If $w(A)/p(A) > w(B)/p(B)$, then $\widetilde{w}(A)/p(A) > \widetilde{w}(B)/p(B)$ because $\sum_{j\in B}\varepsilon^{2j} \leqslant \sum_{j=1}^{n}\varepsilon^{2j} < \varepsilon < 1/p(N)$. Moreover, $\widetilde{w}(A)/p(A) \neq \widetilde{w}(B)/p(B)$ for any two disjoint sets $A, B \subseteq N$ with $w(A)/p(A) = w(B)/p(B)$. Indeed, if, without loss of generality A contains a job k whose index is smaller than that of any job in B, then we add at least $\varepsilon^{2k}/p(N)$ to the ratio of A. On the other hand, we add at most $\sum_{j=k+1}^{n}\varepsilon^{2j} < 2\varepsilon^{2(k+1)} < \varepsilon^{2k}/(2p(N)^2)$ to the ratio of B.

It remains to show that [CH-LP] has a unique optimal solution. We can reuse the proof of Theorem 3.2 for this purpose. In fact, we only need to observe that the non-Sidney-decomposable set S in case (ii) is unique. Hence, by Theorem 3.1, all optimal solutions to [CH-LP] satisfy $\delta_{ij} = 1$ for all $i \in S$, $j \in N\backslash S$. The result follows by induction. □

The perturbation specified in Corollary 3.4 can actually be used for general precedence constraints: afterwards, the Sidney decomposition is unique, and any optimal schedule for the new instance is optimal for the original instance as well.

3.4. Two-dimensional precedence constraints. Two-dimensional partial orders are a generalization of series-parallel partial orders. Dushnik and Miller [6] introduced two-dimensional partial orders in connection with the dimension of a partial order. We refer to Möhring [18] for a survey. A *linear extension* L of a partial order $G = (N, P)$ is a total ordering (acyclic tournament) of the elements in N such that $(i, j) \in L$ for all $(i, j) \in P$. The *dimension* of a partial order $G = (N, P)$ is the smallest number k of linear extensions $L_1, L_2, \ldots, L_k$ whose intersection is P; i.e., $(i, j) \in P$ if and only if $(i, j) \in L_\ell$ for all $\ell = 1, 2, \ldots, k$. Equivalently, P has dimension k if and only if it can be embedded into the k-dimensional Euclidean space where each element $i \in N$ is represented by a point $x^i = (x_1^i, x_2^i, \ldots, x_k^i)$ such that $(i, j) \in P$ if and only if $x_\ell^i < x_\ell^j$ for all $\ell = 1, 2, \ldots, k$ (Ore [21]). We shall need another characterization of two-dimensional partial orders; we provide a proof for completeness. A linear extension L of P is *nonseparating* if $(i, j) \in P$ and $k \parallel \{i, j\}$ imply that either $(k, i) \in L$ or $(j, k) \in L$.

Theorem 3.3 (Dushnik and Miller [6]). *A partial order is two-dimensional if and only if it has a nonseparating linear extension.*

Proof. Let $G = (N, P)$ be a two-dimensional partial order and consider an embedding x into $\mathbb{R}^2$. Let L be the linear ordering over N defined by the values of the first coordinate: $(i, j) \in L$ if and only if $x_1^i < x_1^j$. Clearly, L is a linear extension of P. Moreover, L is nonseparating because if $(i, j) \in P$ and $k \parallel \{i, j\}$, then either $x_1^k < x_1^i$ (and $x_2^j < x_2^k$) or $x_1^j < x_1^k$ (and $x_2^k < x_2^i$), as illustrated in Figure 2.

For the other direction, assume that L is a nonseparating linear extension of P. It is easy to check that $P = L \cap L'$ where $L' := \{(i, j) \in N \times N\colon (i, j) \in P \text{ or } (j, i) \in L\backslash P\}$. It remains to show that L' is a linear ordering; i.e., acyclic. Suppose that L' contains a cycle. If a

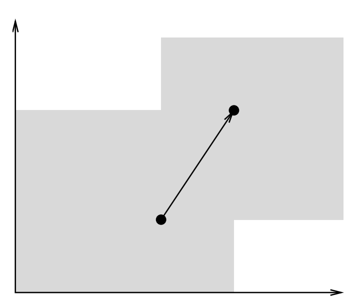

Figure 2. All points that are neither predecessors nor successors of either job in the highlighted precedence constraint are contained in the two non-shaded regions.

tournament contains a cycle, then it contains one with three arcs. Let $(i, j), (j, k), (k, i) \in L'$ be such a cycle. Because L is a linear ordering, at least one of these arcs has to be in P; without loss of generality $(i, j) \in P$. None of the other two arcs can belong to P. Hence, we obtain $(i, k), (k, j), (i, j) \in L$, which is a contradiction because L is nonseparating. □

Two-dimensional partial orders can be recognized in polynomial time (Pnueli et al. [24]), and it is easy to extract a nonseparating linear extension. We will now show that two-dimensional partial orders are precisely the class of partial orders for which the vertex cover graph $G_{CS}(P)$ associated with [CS′-LP] is bipartite. This actually implies that the extreme points of the feasible region of [CS′-LP] are integral if and only if $G = (N, P)$ has dimension 2. First, we need some additional notation. A (not necessarily acyclic) tournament $L \subset N \times N$ is an *extension* of P if $(i, j) \in L$ for all $(i, j) \in P$. It is a *nonseparating* extension if $(i, j) \in P$ and $k \parallel \{i, j\}$ imply that either $(k, i), (k, j) \in L$ or $(i, k), (j, k) \in L$.

Proposition 3.1. *The vertex cover graph $G_{CS}(P)$ associated with* [CS′-LP] *is bipartite if and only if P has a nonseparating extension.*

Proof. Let us assume first that P has a nonseparating extension L. Define $A := L \setminus P$ and $B := \{(i, j): (j, i) \in A\}$. We claim that $G_{CS}(P)$ only has edges between A and B. Recall that there are three different types of edges corresponding to the three different types of inequalities (4b), (4c), and (4d). For the first case, (i, j) and (j, i) are on different sides by definition of A and B. If $(i, j) \in P$ and $k \parallel \{i, j\}$, then the fact that L is nonseparating implies that either $(k, i), (k, j) \in A$ or $(i, k), (j, k) \in A$. This settles the second case. In the third case, we want to show that (i, ℓ) and (k, j) are on different sides of the partition for $(i, j), (k, \ell) \in P$, $i \parallel \ell$, and $j \parallel k$. Because L is nonseparating, either $(i, \ell), (j, \ell) \in A$ or $(\ell, i), (\ell, j) \in A$. Similarly, either $(k, j), (\ell, j) \in A$ or $(j, k), (j, \ell) \in A$. It follows that $(i, \ell) \in A$ implies $(k, j) \in B$ and vice versa. Hence, $G_{CS}(P)$ is bipartite.

On the other hand, if $G_{CS}(P)$ is bipartite, then the nodes (i, j) and (j, i) for $i, j \in N$ with $i \parallel j$ are on different sides of the bipartition because of (4b). Let A and B be the two sides of the bipartition. We define an extension L of P by setting $L := A \cup P$. We claim that L is nonseparating. Indeed, suppose $(i, j) \in P$ and $k \parallel \{i, j\}$. By (4c), (i, k) and (k, j) are on different sides of the bipartition, and so are (k, i) and (j, k). Hence, either $(k, i), (k, j) \in A$ or $(i, k), (j, k) \in A$. □

We follow up with one of the main results of this section.

Theorem 3.4. *The vertex cover graph $G_{CS}(P)$ associated with* [CS′-LP] *is bipartite if and only if P is two-dimensional.*

Proof. With Proposition 3.1 and Theorem 3.3 already in place, we only need to show that if a partial order has a nonseparating extension, then it has a nonseparating linear extension. So, let L be a nonseparating extension of P and assume that L contains a cycle $(i, j), (j, k), (k, i)$. Let $(i)^+ := \{j \in N: (i, j) \in L\}$ and $(i)^- := \{j \in N: (j, i) \in L\}$, as depicted in Figure 3.

It follows from the transitivity of P and from L being nonseparating that no job in $(i)^+$ can be the predecessor of any job in $(i)^-$ with respect to P. Therefore, $L' := (L \setminus ((i)^+ \times (i)^-)) \cup ((i)^- \times (i)^+)$ is an extension of P. Let us check that L' is nonseparating. To this end, consider $(r, s) \in P$ and $t \parallel \{r, s\}$. We have to show that either $(t, r), (t, s) \in L'$ or $(r, t), (s, t) \in L'$. We distinguish three cases:

(i) $r, s \in (i)^+$ or $r, s \in (i)^-$. If $r, s, t \in (i)^+$ or $r, s, t \in (i)^-$, the claim follows because L is nonseparating. If $t \in \{i\} \cup (i)^-$ and $r, s \in (i)^+$, then $(t, r), (t, s) \in L'$, and we are done. Similarly, if $t \in \{i\} \cup (i)^+$ and $r, s \in (i)^-$, then $(r, t), (s, t) \in L'$.

(ii) $r = i$ or $s = i$. In this case there is nothing to prove because L and L' coincide in the arcs adjacent to r and s.

(iii) $r \in (i)^-$ and $s \in (i)^+$. In this case, $(r, i) \in P$ or $(i, s) \in P$ because L is nonseparating. Let us assume that $(r, i) \in P$. (The other case can be handled similarly.) We consider three subcases:

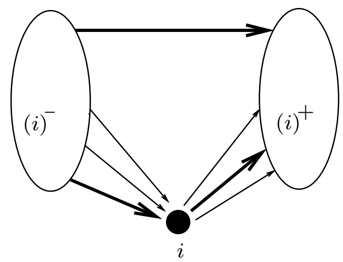

FIGURE 3. Definition of $(i)^+$ and $(i)^-$. Bold arcs represent precedence constraints and cannot go from $(i)^+$ to $(i)^-$. The extension L contains at least one arc from $(i)^+$ to $(i)^-$, forming a cycle. In L', all arcs are directed from $(i)^-$ to $(i)^+$.

(a) $t \in (i)^+$. Suppose that $(r, t), (t, s) \in L'$. Because $(t, s) \in L$ and L is nonseparating, $(r, t) \notin L$. However, then $(i, t), (t, r) \in L$, which together with $(r, i) \in P$ contradicts the fact that L is nonseparating.

(b) $t \in (i)^-$. Suppose that $(r, t), (t, s) \in L'$. This time we can deduce that $(r, t), (t, i) \in L$ contradicting again the nonseparability of L.

(c) $t = i$. This cannot happen because $t \parallel \{r, s\}$.

We infer that L' is nonseparable. Moreover, L' contains at least one less cycle than L. We obtain the result inductively. $\square$

Because it is well known that the feasible region of the vertex cover linear program is integer if and only if the underlying graph is bipartite (Nemhauser and Trotter [19]), we immediately have the following corollary to Theorem 3.4.

COROLLARY 3.5. *The partial order $G = (N, P)$ of an instance of $1|prec|\sum w_j C_j$ is of dimension 2 if and only if every basic feasible solution of the corresponding linear programming problem* [CS-LP] *is integer.*

It is illuminating to go back and consider [CH′-LP] for two-dimensional precedence constraints. Let L be a nonseparating linear extension. If one eliminates all variables δ_{ij} with $(j, i) \in L$ by using the equations $\delta_{ij} + \delta_{ji} = 1$, then the constraints (4c) in the remaining variables are either $\delta_{ki} \leqslant \delta_{kj}$ or $\delta_{jk} \leqslant \delta_{ik}$. Written in this form, it is obvious that [CH′-LP] is a minimum weight closure problem. The *minimum weight closure problem* in a node-weighted digraph is the problem of finding a subset C of nodes of minimum weight such that $v \in C$ for all arcs (u, v) with $u \in C$. Using binary variables z_u, it has the following integer programming formulation:

$$\begin{aligned} \min \quad & \sum_u w_u z_u \\ \text{s.t.} \quad & z_u - z_v \leqslant 0 \quad \text{for all arcs } (u, v), \\ & z_u \in \{0, 1\} \quad \text{for all nodes } u. \end{aligned}$$

It is known that the constraint matrix is totally unimodular. In fact, a simple transformation shows that the minimum weight closure problem is equivalent to the minimum cut problem (Balinski [2], Rhys [29], Picard [22], and Chang and Edmonds [3]).

Let us now turn to the study of approximation algorithms for $1|\text{prec}|\sum w_j C_j$ when the precedence graph is of dimension 2. Kolliopoulos and Steiner [14] presented an approximation algorithm with performance guarantee $(\sqrt{5}+1)/2 + \varepsilon$ for this problem, using machinery developed by Woeginger [34]. Here, we give a simple, combinatorial 3/2-approximation algorithm. It is important to emphasize that the complexity of the scheduling problem with two-dimensional partial orders is still open. Together with the above results, the correctness of Conjecture 2.1 would imply that $1|\text{prec}|\sum w_j C_j$ is solvable in polynomial time for two-dimensional precedence constraints. On the other hand, if this problem is NP-hard, then Conjecture 2.1 is false, unless P = NP!

Our 3/2-approximation algorithm first computes a Sidney decomposition $(S_1, S_2, \ldots, S_k)$. Note that the partial order induced by S_i is also two-dimensional for each $i = 1, 2, \ldots, k$. Let L_i and L_i' be two linear extensions whose intersection is equal to this partial order. For each $i = 1, 2, \ldots, k$, we choose the linear extension from L_i and L_i' that results in the sequence with a smaller objective function value for job set S_i. We create the entire sequence by concatenating the chosen linear extensions in order.

Theorem 3.5. *The scheduling problem $1|prec|\sum w_j C_j$ with two-dimensional precedence constraints has a 3/2-approximation algorithm.*

Proof. We will actually prove a slightly stronger result. Namely, we will show that the above algorithm is a 3/2-approximation algorithm for the $\sum_j w_j S_j$ objective, where $S_j := C_j - p_j$ is the starting time of job j. Theorem 3.1 implies that we can restrict our analysis to the case in which the set N of jobs is ρ-maximal. Let L and L' be two linear extensions such that $P = L \cap L'$. For $X \subset N \times N$, we define $C(X) := \sum_{(i,j)\in X} p_i w_j$. Moreover, we let $C(N) := \sum_{j\in N} w_j p_j$. Observe first that

$$C(L) + C(L') \leqslant w(N)p(N) + C(P) - C(N).$$

Obviously, $C(P) \leqslant \text{OPT}$, where OPT is the weighted sum of starting times of an optimal schedule. In the next section, we will show that the weighted sum of starting times of every feasible schedule of a ρ-maximal instance is at least $w(N)p(N)/2 - C(N)/2$ (Lemma 3.2). Hence, $C(L) + C(L') \leqslant 3\,\text{OPT}$. The result follows. □

Note that one can replace OPT in the above proof with the optimal value of [CH-LP], where the objective function is just $\sum_{i,j\in N} p_i w_j \delta_{ij}$.

3.5. Bounds. We now give a closed-form lower bound on the value of an optimal solution to [CH-LP] for ρ-maximal instances. The bound implies the following lower bound on the value of a feasible schedule, which was used by Chekuri and Motwani [4] and Margot et al. [17] to show that each algorithm in family (C) is a 2-approximation: for a ρ-maximal set N and for any feasible schedule of the jobs in N,

$$w(N)p(N)/2 \leqslant \sum_{j\in N} w_j C_j. \tag{5}$$

Actually, it is not difficult to show (e.g., by using two-dimensional Gantt charts as in Goemans and Williamson [9]) that one can replace $\sum_{j\in N} w_j C_j$ in (5) by the weighted sum of midpoints $\sum_{j\in N} w_j M_j$. Here, $M_j := C_j - p_j/2$. The following lemma provides a slightly stronger result by replacing the midpoints of a feasible schedule with that of any fractional solution.[6]

Lemma 3.2. *Let δ be a feasible solution to* [CS-LP] *over a ρ-maximal ground set N. Then,*

$$w(N)p(N)/2 \leqslant \sum_{i,j\in N} p_i w_j \delta_{ij} + \sum_{j\in N} p_j w_j/2. \tag{6}$$

Proof. Let δ be an optimal basic feasible solution to [CH-LP]. We know from previous arguments (or Chudak and Hochbaum [5, Theorem 2.4]) that δ is half-integral. We can

[6] Theorem 4.5 of Margot et al. [17] yields the same lower bound for the weighted sum of completion times of any feasible solution to [CS-LP].

therefore express the right-hand side of (6) as

$$\sum_{i,\,j\in N} p_i w_j \delta_{ij} + \sum_{j\in N} p_j w_j/2 = \sum_{\substack{i,\,j\in N\\ \delta_{ij}=1}} p_i w_j + \sum_{\substack{i,\,j\in N\\ \delta_{ij}=1/2}} p_i w_j/2 + \sum_{j\in N} p_j w_j/2.$$

On the other hand, we can rephrase the left-hand side as well:

$$w(N)p(N)/2 = \sum_{\substack{i,\,j\in N\\ \delta_{ij}=1}} p_i w_j/2 + \sum_{\substack{i,\,j\in N\\ \delta_{ij}=0}} p_i w_j/2 + \sum_{\substack{i,\,j\in N\\ \delta_{ij}=1/2}} p_i w_j/2 + \sum_{j\in N} p_j w_j/2.$$

After canceling terms, all we have to prove is

$$\sum_{\substack{i,\,j\in N\\ \delta_{ij}=1}} p_i w_j \geqslant \sum_{\substack{i,\,j\in N\\ \delta_{ij}=0}} p_i w_j.$$

It will be helpful to rewrite both terms again:

$$\sum_{\substack{i,\,j\in N\\ \delta_{ij}=1}} p_i w_j = \sum_{j\in N} w_j \sum_{\substack{i\in N\\ \delta_{ij}=1}} p_i \quad \text{and} \quad \sum_{\substack{i,\,j\in N\\ \delta_{ij}=0}} p_i w_j = \sum_{j\in N} w_j \sum_{\substack{i\in N\\ \delta_{ji}=1}} p_i.$$

For a fixed job $j \in N$, the sets $I_j := \{i \in N\colon\ \delta_{ij} = 1\}$ and $F_j := \{i \in N\colon\ \delta_{ji} = 1\}$ are an initial set and a final set in G_N, respectively, because δ is feasible. Lemma 2.1 implies that $\rho(I_j) \leqslant \rho(N) \leqslant \rho(F_j)$, and so we obtain

$$\sum_{\substack{i,\,j\in N\\ \delta_{ij}=1}} p_i w_j \geqslant \frac{1}{\rho(N)} \sum_{\substack{i,\,j\in N\\ \delta_{ij}=1}} w_i w_j \geqslant \sum_{\substack{i,\,j\in N\\ \delta_{ji}=1}} p_i w_j,$$

which concludes the proof. □

One consequence of Lemma 3.2 is a strengthening of the result by Chekuri and Motwani [4] and Margot et al. [17]. Namely, we can show that every schedule produced by Algorithm (C) is within a factor of 2 for the objective of minimizing the weighted sum of *starting times*. Because of Theorem 3.1, we can confine our analysis to ρ-maximal sets N. Let S_j be the starting time of job j in any feasible schedule; C_j is its completion time. Let δ be an optimal solution to [CH-LP]. Then,

$$\sum_{j\in N} w_j S_j = \sum_{j\in N} w_j C_j - \sum_{j\in N} w_j p_j \leqslant w(N)p(N) - \sum_{j\in N} w_j p_j \leqslant 2 \sum_{i,\,j\in N} p_i w_j \delta_{ij}.$$

Here, we used (6) for the second inequality. As $\sum_{i,\,j\in N} p_i w_j \delta_{ij}$ is a lower bound on the optimal weighted sum of starting times, the result follows.

COROLLARY 3.6. *Every schedule that is consistent with Sidney's decomposition is a* 2*-approximation for the weighted sum of starting times objective.*

At the same time, it follows that the optimal value of [CH-LP] (and, therefore, that of [P]) is within a factor 2 of that of the optimal schedule. A construction proposed by Chekuri and Motwani [4] implies that this bound is tight (for [P] and, hence, for [CH-LP]), regardless of whether one considers the weighted sum of starting times or the weighted sum of completion times objective.

4. Structure of the LP in completion time variables. In this section, we study properties of the [QW-LP] relaxation of $1|\text{prec}|\sum w_j C_j$. In particular, we show that Algorithm (A) is consistent with Sidney's decomposition as well. Moreover, we relate the structure of the optimal solutions to this linear programming relaxation to Sidney's decomposition theorem.

We show that in an optimal solution of [QW-LP], the tight parallel inequalities are exactly those corresponding to the sets in a Sidney decomposition. Margot et al. [17] proved this result for the case where [QW-LP] is solved over a non-Sidney-decomposable set.

Borrowing notation from Margot et al. [17], we define the family of tight sets associated with an optimal solution C to [QW-LP] as

$$\tau(C) := \{S \subseteq N\colon \text{ inequality (3b) is tight for } S\}.$$

By convention, $\varnothing \in \tau(C)$. The following lemma combines Lemmas 4.1 and 4.2 in Margot et al. [17].

Lemma 4.1 (Margot et al. [17]). *Let C be an optimal solution to* [QW-LP]. *Let $S \in \tau(C)$ be a nontrivial tight set* (*i.e.,* $S \neq \varnothing$).

(i) *For all jobs $i \in S$, $C_i \leqslant p(S)$, and the inequality holds with equality if and only if $S \backslash \{i\} \in \tau(C)$.*

(ii) *For all jobs $j \notin S$, $C_j \geqslant p(S) + p_j$, and the inequality holds with equality if and only if $S \cup \{j\} \in \tau(C)$.*

(iii) *If there is no tight set $T \subset S$ with $|T| = |S| - 1$, then $C_j - C_i > p_j$ for all $i \in S$, $j \in N \backslash S$.*

(iv) *If there is no tight set $T \supset S$ with $|T| = |S| + 1$, then $C_j - C_i > p_j$ for all $i \in S$, $j \in N \backslash S$.*

Throughout this section, we assume that C is an optimal solution to [QW-LP] and that $C_1 \leqslant C_2 \leqslant \cdots \leqslant C_n$. Parts (i) and (ii) of Lemma 4.1 imply that all sets in $\tau(C)$ are of the form $\{1, \ldots, k\}$ for some $k \in \{0, 1, \ldots, n\}$. For $A, B \subseteq N = \{1, \ldots, n\}$ with $A \cap B = \varnothing$ and $\varepsilon > 0$, we define the perturbed completion time vector $C^\varepsilon(A, B)$ as follows:

$$C_i^\varepsilon(A, B) := \begin{cases} C_i + \varepsilon/p(A) & \text{for all } i \in A, \\ C_i - \varepsilon/p(B) & \text{for all } i \in B, \\ C_i & \text{for all other jobs.} \end{cases}$$

Theorem 4.1. *Let C be an optimal solution to* [QW-LP], *and let $\{S_1, S_2, \ldots, S_k\}$ be the reduced Sidney decomposition of N. Then, inequality* (3b) *is tight for the sets $S_1, S_1 \cup S_2, \ldots, S_1 \cup S_2 \cup \cdots \cup S_k$.*

Proof. Suppose that the claim is false. Let ℓ be the smallest index in $\{1, 2, \ldots, k-1\}$[7] such that $Q_\ell := S_1 \cup S_2 \cup \cdots \cup S_\ell \notin \tau(C)$, and let $\{1, \ldots, r\}$ and $\{1, \ldots, s\}$ be the largest set in $\tau(C)$ contained in Q_ℓ and the smallest set in $\tau(C)$ containing Q_ℓ, respectively. We define $C^\varepsilon(A, B)$ by taking $A := (N \backslash Q_\ell) \cap \{r+1, \ldots, s\}$ and $B := Q_\ell \cap \{r+1, \ldots, s\}$. Let us argue that $C^\varepsilon(A, B)$ is a feasible solution of [QW-LP]. As $\{1, \ldots, q\} \notin \tau(C)$ for $r + 1 \leqslant q \leqslant s - 1$, $C^\varepsilon(A, B)$ satisfies all parallel inequalities (3b) for sufficiently small $\varepsilon > 0$. Now, consider the precedence constraint (3c) for $(i, j) \in P$. If $i \in A$ but $j \notin A$, then $j > s$. As $\{1, \ldots, s\} \in \tau(C)$ but $\{1, \ldots, s-1\} \notin \tau(C)$, it follows from Lemma 4.1(iii) that $C_j - C_i > p_j$. If $j \in B$ but $i \notin B$, then $i \in \{1, \ldots, r\}$. Because $\{1, \ldots, r\} \in \tau(C)$ but $\{1, \ldots, r+1\} \notin \tau(C)$, it follows from Lemma 4.1 (iv) that $C_j - C_i > p_j$. In either case, $C^\varepsilon(A, B)$ is a feasible solution to [QW-LP] for sufficiently small $\varepsilon > 0$. Because A is initial in $G_{N \backslash Q_\ell}$ and B is final in G_{S_ℓ}, the difference in the objective function values between C and $C^\varepsilon(A, B)$ is $\varepsilon(\rho(B) - \rho(A)) > 0$. This contradicts the optimality of C; consequently, $Q_\ell \in \tau(C)$ for $\ell = 1, 2, \ldots, k$. $\square$

By applying parts (i) and (ii) of Lemma 4.1 to the tight sets specified in Theorem 4.1, we obtain the main result of this section.

[7] Inequality (3b) is always tight for $N = S_1 \cup S_2 \cup \cdots \cup S_k$ because we assumed that $w(F) > 0$ for all final sets $F \subseteq N$.

COROLLARY 4.1. *Let C be an optimal solution to* [QW-LP], *and let $\{S_1, S_2, \ldots, S_k\}$ be the reduced Sidney decomposition of N. Then, $C_j - C_i \geqslant p_j$ for all pairs i, j of jobs with $i \in S_\ell$ and $j \in S_{\ell+1}$ for some $\ell \in \{1, 2, \ldots, k-1\}$. In particular, Algorithm* (A) *is consistent with Sidney's decomposition.*

Let us finally show that the sets in a Sidney decomposition are essentially the only tight sets in an optimal solution to [QW-LP]. For this, we have to look at an appropriate Sidney decomposition, which we get by assuming that C is the unique optimal solution.[8] Let $\{Q_1, Q_2\backslash Q_1, \ldots, Q_k\backslash(Q_1 \cup \ldots \cup Q_{k-1})\}$ be a Sidney decomposition of N such that $Q_i = \{1, \ldots, q_i\}$ with $q_1 < q_2 < \cdots < q_k = n$. Suppose that $\{1, \ldots, r\} \in \tau(C)$ for some $q_i < r < q_{i+1}$. If, in addition, r is such that $\{1, \ldots, r+1\} \notin \tau(C)$, then we let $A := \{q_i + 1, \ldots, r\}$ and $B := \{r+1, \ldots, q_{i+1}\}$. Consider $C^\varepsilon(A, B)$. Using Lemma 4.1, one can easily check that $C^\varepsilon(A, B)$ is feasible. Note that A is initial in $G_{q_i+1, \ldots, q_{i+1}}$ while B is final in $G_{q_i+1, \ldots, q_{i+1}}$. Therefore, the objective function value of $C^\varepsilon(A, B)$ is not worse than that of C—a contradiction. Thus, all parallel inequalities associated with the sets $\{1, \ldots, s\}$ must be tight for $q_i \leqslant s \leqslant q_{i+1}$. Therefore,

$$C_j = C_{q_i} + \sum_{\ell=q_i+1}^{j} p_\ell \quad \text{for all } q_i \leqslant j \leqslant q_{i+1}.$$

This implies that $Q_{i+1}\backslash Q_i$ must form a chain; i.e., $(q_i + 1, q_i + 2)$, $(q_i + 2, q_i + 3)$, $\ldots$, $(q_{i+1} - 1, q_{i+1}) \in P$. Otherwise, $C^\varepsilon(A, B)$ with $A := \{q_i + 1, \ldots, t\}$ and $B := \{t + 1, \ldots, q_{i+1}\}$ for t such that $(t, t+1) \notin P$, is another optimal solution for [QW-LP]. It follows that inequalities (3b) for $\{1, \ldots, r\}$ for $q_i < r < q_{i+1}$ are implied by the parallel inequalities for the sets in a Sidney decomposition and inequalities (3c). We have proved the following theorem, which characterizes the tight inequalities of an optimal solution to [QW-LP].

THEOREM 4.2. *Let C be the unique optimal solution to* [QW-LP], *and let $\{S_1, S_2, \ldots, S_k\}$ be a Sidney decomposition of N. Then, the parallel inequalities* (3b) *are tight for the sets $S_1, S_1 \cup S_2, \ldots, S_1 \cup S_2 \cup \cdots \cup S_k$. All other parallel inequalities* (3b) *are either redundant or not tight.*

Acknowledgments. The authors are grateful to an anonymous referee whose suggestion led to the current, simplified version of the proof of Theorem 3.5. They also thank both anonymous referees for their useful comments, which helped to improve the presentation of this paper.

References

[1] Balas, E. 1985. On the facial structure of scheduling polyhedra. *Math. Programming Stud.* **24** 179–218.
[2] Balinski, M. 1970. On a selection problem. *Management Sci.* **17** 230–231.
[3] Chang, G. J., J. Edmonds. 1985. The poset scheduling problem. *Order* **2** 113–118.
[4] Chekuri, C., R. Motwani. 1999. Precedence constrained scheduling to minimize sum of weighted completion time on a single machine. *Discrete Appl. Math.* **98** 29–38.
[5] Chudak, F. A., D. S. Hochbaum. 1999. A half-integral linear programming relaxation for scheduling precedence-constrained jobs on a single machine. *Oper. Res. Lett.* **25** 199–204.
[6] Dushnik, B., E. W. Miller. 1941. Partially ordered sets. *Amer. J. Math.* **63** 600–610.
[7] Dyer, M. E., L. A. Wolsey. 1990. Formulating the single machine sequencing problem with release dates as a mixed integer program. *Discrete Appl. Math.* **26** 255–270.
[8] Eastman, W. L., S. Even, I. Isaacs. 1964. Bounds for the optimal scheduling of n jobs on m processors. *Management Sci.* **11** 268–279.
[9] Goemans, M. X., D. P. Williamson. 2000. Two-dimensional Gantt charts and a scheduling algorithm of Lawler. *SIAM J. Discrete Math.* **13** 281–294.

[8] If $C_1 \leqslant C_2 \leqslant \cdots \leqslant C_n$, one can replace the original job weights w_j with $\widetilde{w}_j := w_j + \varepsilon^{2j}$ for some sufficiently small $\varepsilon > 0$. Then, C is the unique optimum, and the Sidney decomposition is unique as well.

[10] Graham, R. L., E. L. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan. 1979. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Ann. Discrete Math.* **5** 287–326.
[11] Hall, L. A., D. B. Shmoys, J. Wein. 1996. Scheduling to minimize average completion time: Off-line and on-line algorithms. *Proc. 7th Annual ACM-SIAM Sympos. on Discrete Algorithms*, Atlanta, GA. SIAM, Philadelphia, PA, 142–151.
[12] Hall, L. A., A. S. Schulz, D. B. Shmoys, J. Wein. 1997. Scheduling to minimize average completion time: Off-line and on-line approximation algorithms. *Math. Oper. Res.* **22** 513–544.
[13] Hochbaum, D. S. 1983. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Appl. Math.* **6** 243–254.
[14] Kolliopoulos, S. G., G. Steiner. 2002. Partially-ordered knapsack and applications to scheduling. R. H. Möhring, R. Raman, eds. *Algorithms–ESA 2002*, Vol. 2461. *Lecture Notes in Computer Science*. Springer, Heidelberg, Germany, 612–624.
[15] Lawler, E. L. 1978. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Ann. Discrete Math.* **2** 75–90.
[16] Lenstra, J. K., A. H. G. Rinnooy Kan. 1978. Complexity of scheduling under precedence constraints. *Oper. Res.* **26** 22–35.
[17] Margot, F., M. Queyranne, Y. Wang. 2003. Decompositions, network flows, and a precedence constrained single-machine scheduling problem. *Oper. Res.* **51** 981–992.
[18] Möhring, R. H. 1989. Computationally tractable classes of ordered sets. I. Rival, ed. *Algorithms and Order*, Kluwer Academic, Dordrecht, The Netherlands, 105–193.
[19] Nemhauser, G. L., L. E. Trotter, Jr. 1974. Properties of vertex packing and independence system polyhedra. *Math. Programming* **6** 48–61.
[20] Nemhauser, G. L., L. E. Trotter, Jr. 1975. Vertex packing: Structural properties and algorithms. *Math. Programming* **8** 232–248.
[21] Ore, O. 1962. *Theory of Graphs*, Vol. XXXVIII, *Colloquium Publications*. *American Mathematical Society*, Providence, RI.
[22] Picard, J.-C. 1976. Maximal closure of a graph and applications to combinatorial problems. *Management Sci.* **22** 1268–1272.
[23] Picard, J.-C., M. Queyranne. 1980. On the structure of all minimum cuts in a network and applications. *Math. Programming Stud.* **13** 8–16.
[24] Pnueli, A., A. Lempel, S. Even. 1971. Transitive orientation of graphs and identification of permutation graphs. *Canadian J. Math.* **23** 160–175.
[25] Potts, C. N. 1980. An algorithm for the single machine sequencing problem with precedence constraints. *Math. Programming Stud.* **13** 78–87.
[26] Queyranne, M. 1993. Structure of a simple scheduling polyhedron. *Math. Programming* **58** 263–285.
[27] Queyranne, M., A. S. Schulz. 1994. Polyhedral approaches to machine scheduling. Technical report 408/1994, Department of Mathematics, Technische Universität Berlin, Berlin, Germany.
[28] Queyranne, M., Y. Wang. 1991. Single-machine scheduling polyhedra with precedence constraints. *Math. Oper. Res.* **16** 1–20.
[29] Rhys, J. M. W. 1970. A selection problem of shared fixed costs and network flows. *Management Sci.* **17** 200–207.
[30] Schulz, A. S. 1996. Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds. W. H. Cunningham, S. T. McCormick, M. Queyranne, eds. *Integer Programming and Combinatorial Optimization*, Vol. 1084. *Lecture Notes in Computer Science*, Springer, Heidelberg, Germany, 301–315.
[31] Schulz, A. S., M. Skutella. 1997. Random-based scheduling: New approximations and LP lower bounds. J. D. P. Rolim, ed. *Randomization and Approximation Techniques in Computer Science*, Vol. 1269. *Lecture Notes in Computer Science*, Springer, Heidelberg, Germany, 119–133.
[32] Sidney, J. B. 1975. Decomposition algorithms for single machine scheduling with precedence relations and deferral costs. *Oper. Res.* **23** 283–298.
[33] Smith, W. E. 1956. Various optimizers for single-stage production. *Naval Res. Logist. Quart.* **3** 59–66.
[34] Woeginger, G. J. 2003. On the approximability of average completion time scheduling under precedence constraints. *Discrete Appl. Math.* **131** 237–252.
[35] Wolsey, L. A. 1985. Mixed integer programming formulations for production planning and scheduling problems. Invited talk at the 12th Internat. Sympos. Math. Programming, MIT, Cambridge, MA.
[36] Wolsey, L. A. 1990. Formulating single machine scheduling problems with precedence constraints. J. J. Gabszewicz, J. F. Richard, L. A. Wolsey, eds. *Economic Decision-Making: Games, Econometrics and Optimisation*, North-Holland, Amsterdam, The Netherlands, 473–484.