MATHEMATICS OF OPERATIONS RESEARCH
Vol. 36, No. 4, November 2011, pp. 653–669
ISSN 0364-765X | EISSN 1526-5471 | 11 | 3604 | 0653
http://dx.doi.org/10.1287/moor.1110.0512

On the Approximability of Single-Machine Scheduling with Precedence Constraints

Christoph Ambühl
Department of Computer Science, The University of Liverpool, Liverpool L69 3BX, United Kingdom

Monaldo Mastrolilli
Instituto Dalle Molle di Studi sull'Intelligenza Artificiale, 6928 Manno-Lugano, Switzerland,
monaldo@idsia.ch, http://www.idsia.ch/~monaldo/

Nikolaus Mutsanas
Université Libre de Bruxelles, Département de Mathématique, 1150 Ixelles-Bruxelles, Belgium,
nikolaus.mutsanas@ulb.ac.be, http://www.idsia.ch/~nikos/

Ola Svensson
School of Computer and Communication Sciences, École Polytechnique Fédérale de Lausanne, CH-1015 Lausanne, Switzerland,
ola.svensson@epfl.ch, http://algo.epfl.ch/~osven/

We consider the single-machine scheduling problem to minimize the weighted sum of completion times under precedence constraints. In a series of recent papers, it was established that this scheduling problem is a special case of minimum weighted vertex cover.

In this paper, we show that the vertex cover graph associated with the scheduling problem is exactly the graph of incomparable pairs defined in the dimension theory of partial orders. Exploiting this relationship allows us to present a framework for obtaining $(2-2/f)$-approximation algorithms, provided that the set of precedence constraints has fractional dimension of at most f. Our approach yields the best-known approximation ratios for all previously considered special classes of precedence constraints, and it provides the first results for bounded degree and orders of interval dimension 2.

On the negative side, we show that the addressed problem remains NP-hard even when restricted to the special case of interval orders. Furthermore, we prove that the general problem, if a fixed cost present in all feasible schedules is ignored, becomes as hard to approximate as vertex cover. We conclude by giving the first inapproximability result for this problem, showing under a widely believed assumption that it does not admit a polynomial-time approximation scheme.

Key words: single-machine scheduling; vertex cover; fractional dimension
MSC2000 subject classification: Primary: 90B35, 68W25; secondary: 05C15, 06A07
OR/MS subject classification: Primary: production/scheduling, approximation/heuristic; secondary: mathematics, combinatorics
History: Received July 18, 2008; revised June 28, 2010, and January 31, 2011. Published online in *Articles in Advance* October 14, 2011.

1. Introduction. The problem we consider in this paper is a classical problem in scheduling theory, known as $1\,|\,\text{prec}\,|\sum w_jC_j$ in standard scheduling notation (see, e.g., Graham et al. [16]). It is defined as the problem of scheduling a set $N=\{1,\ldots,n\}$ of n jobs on a single machine that can process at most one job at a time. Each job j has a processing time p_j and a weight w_j, where p_j and w_j are nonnegative integers. Jobs also have precedence constraints between them that are specified in the form of a *partially ordered set* (*poset*) $\mathbf{P}=(N,P)$, consisting of the set of jobs N and a partial order, i.e., a reflexive, antisymmetric, and transitive binary relation P on N, where $(i,j)\in P$ $(i\neq j)$ implies that job i must be completed before job j can be started. The goal is to find a non-pre-emptive schedule that minimizes $\sum_{j=1}^n w_jC_j$, where C_j is the time at which job j completes in the given schedule.

The described problem was already shown to be strongly NP-hard in 1978 by Lawler [23] and Lenstra and Rinnooy Kan [25]. For the general version of $1\,|\,\text{prec}\,|\sum w_jC_j$, closing the approximability gap has been listed as one of ten outstanding open problems in scheduling theory (see, e.g., Schuurman and Woeginger [36]). On the positive side, several polynomial-time 2-approximation algorithms are known. Pisaruk [31] claims to have obtained the first such algorithm. Schulz [35] and Hall et al. [18] gave 2-approximation algorithms using linear programming relaxations.[1] Chudak and Hochbaum [9] gave another algorithm based on a relaxation of the linear program studied by Potts [32]. Independently, Chekuri and Motwani [8] and Margot et al. [27] provided identical, extremely simple 2-approximation algorithms based on Sidney's decomposition theorem (Sidney [37]). Correa and Schulz [10] proved that all known 2-approximation algorithms follow a Sidney decomposition and therefore belong to the class of approximation algorithms described by Chekuri and Motwani [8] and Margot et al. [27].

[1] Note that Hall et al. [18] is a joint journal version of Schulz [35] and Hall et al. [17] in which the latter authors give a $(4-\epsilon)$ approximation algorithm for this problem.

A close inspection of the objective function reveals that it consists of a *fixed cost* (i.e., a cost that only depends on the instance and is present in any feasible solution) and a *variable cost*. It turns out that the fixed cost of the objective function is crucial for the 2-approximation guarantee of the above-mentioned class of algorithms. Indeed, in his dissertation, Uhan [43] gave an instance for which any solution following the Sidney decomposition performs arbitrarily bad for the variable cost.

On the negative side, Woeginger [44] proved that many quite severe restrictions on the weights and processing times do not influence approximability. For example, the special case in which all jobs either have $p_j = 1$ and $w_j = 0$ or $p_j = 0$ and $w_j = 1$ is as hard to approximate as the general case. On the other hand, imposing restrictions on the partial order can lead to better approximation or even exact algorithms. Indeed, Lawler [23] already gave an exact algorithm for series-parallel orders in 1978 (other classes of polynomially solvable instances are also known and the interested reader is referred to Lawler et al. [24] for a survey). For interval orders and convex bipartite precedence constraints, Woeginger [44] gave approximation algorithms with approximation ratio arbitrarily close to the golden ratio $\frac{1}{2}(1+\sqrt{5}) \approx 1.61803$.

Recently, Ambühl and Mastrolilli [2] settled an open problem first raised by Chudak and Hochbaum [9] and whose answer was subsequently conjectured by Correa and Schulz [10]. Correa and Schulz [10] show that a positive answer of an open question raised in Chudak and Hochbaum [9] would imply that $1|\text{prec}|\sum w_j C_j$ is a special case of the weighted vertex cover problem, and that 2-dimensional orders would be solvable in polynomial time (thus improving upon their 3/2-approximation (Correa and Schulz [10]), which, in turn, improved upon a previous approximation algorithm with a guarantee of $(1+\sqrt{5})/2+\varepsilon$; see Kolliopoulos and Steiner [22]). More precisely, the results in Correa and Schulz [10] and in Ambühl and Mastrolilli [2] prove that every instance S of $1|\text{prec}|\sum w_j C_j$ can be translated in polynomial time into a weighted graph $G^S_{\mathbf{P}}$, such that finding an optimum of S can be reduced to finding an optimum vertex cover in $G^S_{\mathbf{P}}$. This result even holds for approximate solutions. Finding an α-approximate solution for S can be reduced to finding an α-approximate vertex cover in $G^S_{\mathbf{P}}$. By using this relationship, several previous results for the scheduling problem can be explained (and, in some cases, improved) by means of the vertex cover theory. Furthermore, the implied 2-approximation algorithm (by approximating vertex cover) is novel in that it already provides an approximation guarantee of 2 for the variable cost in contrast to the previously known algorithms mentioned earlier.

In this paper, we continue to investigate the structure of $1|\text{prec}|\sum w_j C_j$. We point out an interesting relationship between the dimension theory of partial orders and $1|\text{prec}|\sum w_j C_j$. More specifically, in §3, we show that the vertex cover graph $G^S_{\mathbf{P}}$ associated with $1|\text{prec}|\sum w_j C_j$ is exactly the graph of incomparable pairs $G_{\mathbf{P}}$ defined in dimension theory (Felsner and Trotter [13]). This equivalence allows us to benefit from dimension theory. In particular, the chromatic number of $G^S_{\mathbf{P}}$ is at most d whenever the dimension of the poset at hand is (at most) d. Hochbaum [20] showed that if a given graph for the vertex cover problem can be colored with d colors in polynomial time, then there exists a $(2-2/d)$-approximation algorithm for the corresponding weighted vertex cover problem. It follows that there exists a $(2-2/d)$-approximation algorithm for $1|\text{prec}|\sum w_j C_j$ for all those special classes of precedence constraints that admit a polynomial-time computable d-realizer.

By following this general approach, we obtain approximation algorithms for relevant special classes of precedence constraints of low dimension such as *convex bipartite* precedence constraints (§4.1) and *semiorders* (§4.2) for which we exhibit 4/3-approximation algorithms that improve previous results by Woeginger [44].

Unfortunately, the framework described here fails in the case of interval orders (in this case, the dimension can be of order $\log\log n$; see Trotter [41]). To overcome this difficulty, we further generalize this framework such that it can be applied to precedence constraints of low *fractional* dimension (Brightwell and Scheinerman [7]) (see §5). The extended framework yields $(2-2/f)$-approximation algorithms whenever precedence constraints have fractional dimension bounded by a constant f and satisfy an additional mild condition (see §5). Because the fractional dimension of interval orders is bounded by 4 (see §6.1), this gives a 3/2-approximation algorithm and improves the previous result in Woeginger [44]. The extended framework can also be applied to posets of interval dimension 2 (see §6.2), bounded degree posets (see §6.3), and posets obtained by lexicographic sums (see §6.4).

In summary, these results indicate a strong relationship between the approximability of $1|\text{prec}|\sum w_j C_j$ and the fractional dimension f of the precedence constraints. In particular, $1|\text{prec}|\sum w_j C_j$ is polynomially solvable for $f = 2$ (Correa and Schulz [10], Ambühl and Mastrolilli [2]) but is already NP-hard for $f \geq 3$. The latter stems from the facts that problem $1|\text{prec}|\sum w_j C_j$ is strongly NP-hard even for posets with in-degree 2 (Lenstra and Rinnooy Kan [25]), and the fractional dimension of these posets is bounded by 3 (Felsner and Trotter [12]). This leaves the complexity for $2 < f < 3$ as an open question.

In the second part of this paper, we present some negative results for this problem. In §7, we show that the addressed problem remains NP-hard even when restricted to the special case of interval orders. This result is rather unexpected as many problems can be solved in polynomial time when restricted to interval orders (see, e.g., Papadimitriou and Yannakakis [30]). The reduction relies heavily on the connection between $1 \mid \text{prec} \mid \sum w_j C_j$ and the weighted vertex cover described in Correa and Schulz [10] and Ambühl and Mastrolilli [2].

For the general problem $1 \mid \text{prec} \mid \sum w_j C_j$ with arbitrary partial orders, we show in §8 that approximating the variable cost of $1 \mid \text{prec} \mid \sum w_j C_j$ is equivalent to approximating the vertex cover problem. This implies that a better than 2-approximation algorithm for $1 \mid \text{prec} \mid \sum w_j C_j$ would need to either use the fixed cost of the objective function or improve the best-known approximation algorithm for the vertex cover problem.

We conclude our work by presenting the first inapproximability result for $1 \mid \text{prec} \mid \sum w_j C_j$ by ruling out, under a fairly standard assumption, the existence of a polynomial-time approximation scheme (PTAS) for the addressed scheduling problem. More precisely, in §9, we establish a connection between $1 \mid \text{prec} \mid \sum w_j C_j$ and the maximum edge biclique problem, which proves that this problem does not admit a PTAS unless all problems in the complexity class NP can be solved by probabilistic algorithms of subexponential running time. This result makes a first step toward closing the approximability gap for this scheduling problem, a prominent problem in scheduling theory (see Schuurman and Woeginger [36]). Subsequent to our work, Bansal and Khot [6] showed that the gap indeed closes, assuming a variant of the unique games conjecture (Khot [21]), by providing a 2-inapproximability result based on that assumption. Furthermore, this year, the fourth author (Svensson [40]) established an interesting relationship between $1 \mid \text{prec} \mid \sum w_j C_j$ and the problem $P \mid \text{prec} \mid C_{\max}$ by showing that a $(2-\epsilon)$-hardness for $1 \mid \text{prec} \mid \sum w_j C_j$ would imply essentially the same hardness result for $P \mid \text{prec} \mid C_{\max}$ and would thus settle another prominent open question listed in Schuurman and Woeginger [36]. This underscores the importance of understanding the approximability of $1 \mid \text{prec} \mid \sum w_j C_j$.

2. Definitions and preliminaries.

2.1. Posets and fractional dimension. Let $\mathbf{P} = (N, P)$ be a poset. For $x, y \in N$, we write $x \leq y$ when $(x, y) \in P$ and $x < y$ when $(x, y) \in P$ and $x \neq y$. When neither $(x, y) \in P$ nor $(y, x) \in P$, we say that x and y are incomparable (denoted by $x \parallel y$). We call $\text{inc}(\mathbf{P}) = \{(x, y) \in N \times N: x \parallel y \text{ in } P\}$ the set of *incomparable pairs* of $\mathbf{P}$ (note that, because incomparability is a symmetric relation, if $(x, y) \in \text{inc}(\mathbf{P})$, then $(y, x) \in \text{inc}(\mathbf{P})$ as well). A poset $\mathbf{P}$ is a *linear order* (or a *total order*) if, for any $x, y \in N$, either $(x, y) \in P$ or $(y, x) \in P$, i.e., $\text{inc}(\mathbf{P}) = \varnothing$. A partial order P' on N is an *extension* of a partial order P on the same set N if $P \subseteq P'$. An extension of $\mathbf{P}$ that is a linear order is called a *linear extension* of $\mathbf{P}$. Mirroring the definition of the fractional chromatic number of a graph, Brightwell and Scheinerman [7] introduce the notion of fractional dimension of a poset. Let $\mathcal{F} = \{L_1, L_2, \ldots, L_t\}$ be a nonempty multiset of linear extensions of $\mathbf{P}$. Brightwell and Scheinerman [7] call $\mathcal{F}$ a *k-fold realizer* of $\mathbf{P}$ if, for each incomparable pair (x, y), there are at least k linear extensions in $\mathcal{F}$ that reverse the pair (x, y), i.e., $|\{i = 1, \ldots, T: y < x \text{ in } L_i\}| \geq k$. We call a k-fold realizer of size t a *k:t-realizer*. We will sometimes abbreviate a 1-fold realizer of size k by *k-realizer*. We call an incomparable pair $(x, y) \in \text{inc}(\mathbf{P})$ a *critical pair* if, for all $z, w \in N \backslash \{x, y\}$ $z < x$ in P implies $z < y$ in P and $y < w$ in P implies $x < w$ in P. Critical pairs play an important role in dimension theory: If, for each critical pair (x, y), there are at least k linear extensions in $\mathcal{F}$ that reverse the pair (x, y), then $\mathcal{F}$ is a k-fold realizer of P and vice versa (Brightwell and Scheinerman [7]). The *fractional dimension of* $\mathbf{P}$ is the smallest rational number $\text{fdim}(\mathbf{P}) \geq 1$ for which there exists a k:t-realizer of $\mathbf{P}$ so that $t/k \leq \text{fdim}(\mathbf{P})$. Using this terminology, the *dimension* of $\mathbf{P}$, denoted by $\dim(\mathbf{P})$, is the least t for which there exists a 1-fold realizer of P. It is immediate that $\text{fdim}(\mathbf{P}) \leq \dim(\mathbf{P})$ for any poset $\mathbf{P}$. Furthermore, $\text{fdim}(\mathbf{P}) = 1$ or $\text{fdim}(\mathbf{P}) \geq 2$ (Brightwell and Scheinerman [7]).

2.2. The scheduling problem and vertex cover. In a series of recent papers (Chudak and Hochbaum [9], Correa and Schulz [10], Ambühl and Mastrolilli [2]), it was proved that $1 \mid \text{prec} \mid \sum w_j C_j$ is a special case of Minimum Weighted Vertex Cover. Given a graph $G = (V, E)$ with weights w_i on the vertices, find a subset $V' \subseteq V$, minimizing the objective function $\sum_{i \in V'} w_i$ such that, for each edge $(u, v) \in E$, at least one of u and v belongs to V'.

This result was achieved by investigating the relationship between several different linear programming formulations and relaxations (Potts [32], Chudak and Hochbaum [9], Correa and Schulz [10]) of $1 \mid \text{prec} \mid \sum w_j C_j$

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. Correa and Schulz [10] proposed the following relaxation of $1\,|\,\text{prec}\,|\sum w_j C_j$:

$$
\begin{aligned}
\text{[CS-IP]} \qquad & \min \ \sum_{i\|j} \delta_{ij} p_i w_j + \sum_{j\in N} p_j w_j + \sum_{(i,j)\in P} p_i w_j \\
& \text{s.t.} \ \ \delta_{ij} + \delta_{ji} \geq 1 \quad i \,\|\, j, && (1)\\
& \qquad \delta_{ik} + \delta_{kj} \geq 1 \quad (i,j) \in P,\ i \,\|\, k, \text{ and } k \,\|\, j, && (2)\\
& \qquad \delta_{i\ell} + \delta_{kj} \geq 1 \quad (i,j), (k,\ell) \in P,\ i \,\|\, \ell, \text{ and } j \,\|\, k, && (3)\\
& \qquad \delta_{ij} \in \{0,1\} \quad i \,\|\, j.
\end{aligned}
$$

Note that [CS-IP] can be interpreted as the problem of finding a minimum weighted vertex cover in an undirected graph (Correa and Schulz [10]): Given a scheduling instance S with precedence constraints $\mathbf{P} = (N, P)$, let[2] $G_{\mathbf{P}}^S$ be the graph that has a node for each incomparable pair (i, j) of jobs weighted by $p_i w_j$. 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 $(i, \ell), (k, j) \in P$. We will denote by [CS-LP] the linear relaxation of [CS-IP].

Correa and Schulz [10] conjectured that an optimal solution to $1\,|\,\text{prec}\,|\sum w_j C_j$ gives an optimal solution to [CS-IP] as well. The conjecture in Correa and Schulz [10] was recently settled in the affirmative by Ambühl and Mastrolilli [2], even for approximate solutions. To summarize, the following theorem was proven.

THEOREM 2.1 (AMBÜHL AND MASTROLILLI [2], CORREA AND SCHULZ [10]). *Let S be an instance of $1\,|\,\text{prec}\,|\sum w_j C_j$, and let $G_{\mathbf{P}}^S$ be the corresponding vertex cover graph. An α-approximate solution to $G_{\mathbf{P}}^S$ can in polynomial time be turned into an α-approximate solution to S.*

It follows that problem $1\,|\,\text{prec}\,|\sum w_j C_j$ is a special case of the weighted vertex cover problem in the graph $G_{\mathbf{P}}^S$. We refer the interested reader to Correa and Schulz [10] and Ambühl and Mastrolilli [2] for more comprehensive discussions.

We already mentioned that Hochbaum [20] gave a $(2-2/k)$-approximation algorithm for the weighted vertex cover problem whenever the vertex cover graph is k-colorable in polynomial time. We summarize at this point with the following observation.

OBSERVATION 2.2 (AMBÜHL AND MASTROLILLI [2], CORREA AND SCHULZ [10], HOCHBAUM [20]). If the graph $G_{\mathbf{P}}^S$ can be colored with k colors in polynomial time, then the problem $1\,|\,\text{prec}\,|\sum w_j C_j$ has a polynomial-time $(2-2/k)$-approximation algorithm.

We will give examples of precedence constraints for which $G_{\mathbf{P}}^S$ can be colored with few colors after we show that the study of the dimension of partially ordered sets can help identify such classes of precedence constraints.

3. Scheduling and dimension theory.

The aim of this section is to point out the connection between $1\,|\,\text{prec}\,|\sum w_j C_j$ and the dimension theory of partial orders. For this purpose, we need some preliminary definitions.

Let $\mathbf{P} = (N, P)$ be any poset that is not a linear order. Felsner and Trotter [13] associate with $\mathbf{P}$ a hypergraph $\mathscr{H}_{\mathbf{P}}$, called the *hypergraph of incomparable pairs* and defined as follows. The vertices of $\mathscr{H}_{\mathbf{P}}$ are the incomparable pairs in $\mathbf{P}$. The edge set consists of those minimal (in terms of inclusion) sets U of incomparable pairs such that no linear extension of $\mathbf{P}$ reverses all incomparable pairs in U. Let $G_{\mathbf{P}}$ denote the ordinary graph, called the *graph of incomparable pairs* and determined by all edges of size 2 in $\mathscr{H}_{\mathbf{P}}$. We recall that the *chromatic number* of a hypergraph $\mathscr{H} = (V, \mathscr{E})$, denoted $\chi(\mathscr{H})$, is the least positive integer t for which there is a function $F\colon V \to \{1, \ldots, t\}$ so that there is no $\alpha \in \{1, \ldots, t\}$ for which there is an edge $E \in \mathscr{E}$ with $f(x) = \alpha$ for every $x \in E$. The following result by Felsner and Trotter [13] associates a poset $\mathbf{P}$ with $\mathscr{H}_{\mathbf{P}}$ so that the dimension of $\mathbf{P}$ is the chromatic number of $\mathscr{H}_{\mathbf{P}}$.

PROPOSITION 3.1 (FELSNER AND TROTTER [13]). *Let $\mathbf{P} = (N, P)$ be a poset that is not a linear order. Then,* $\dim(\mathbf{P}) = \chi(\mathscr{H}_{\mathbf{P}}) \geq \chi(G_{\mathbf{P}})$.

[2] Note that, though the poset $\mathbf{P}$ is part of the definition of S and could therefore be omitted, we use this notation to adhere to the notation of the graph of incomparable pairs in dimension theory (see §3). Thus, $\mathbf{P}$ defines the *structure* of the graph $G_{\mathbf{P}}^S$, and the rest of S defines the *weights* of its vertices.

Given a k-realizer $\mathcal{R} = \{L_1, L_2, \ldots, L_k\}$ of $\mathbf{P}$, we can easily color $\mathcal{H}_\mathbf{P}$ (and $G_\mathbf{P}$) with k colors: Color vertex (i, j) with some color c whenever $(j, i) \in L_c$ (if (j, i) appears in several linear extensions, pick one arbitrarily). Observe that, if all nodes of a hyperedge U are colored by the same color c, then the linear extension L_c reverses all incomparable pairs of U, which is impossible by the definition of $\mathcal{H}_\mathbf{P}$. Brightwell and Scheinerman [7] noted that the same relationship holds for the fractional versions, i.e., $\operatorname{fdim}(\mathbf{P}) = \chi_f(\mathcal{H}_\mathbf{P}) \geq \chi_f(G_\mathbf{P})$, where $\chi_f(A)$ denotes the fractional chromatic number of A. We refer the reader to Scheinerman and Ullman [34] for an introduction to fractional graph coloring.

The following proposition is immediate and can be easily checked. It furthers the relationship between dimension theory and the approximability of $1|\text{prec}|\sum w_j C_j$.

Proposition 3.2. *The vertex cover graph $G^S_\mathbf{P}$ associated with $1|\text{prec}|\sum w_j C_j$ and the graph of incomparable pairs $G_\mathbf{P}$ coincide.*

The combinatorial theory of partially ordered sets is a well-studied field. Tapping this source can help in designing approximation algorithms. The following theorem is such an example.

Theorem 3.1 (Felsner and Trotter [13], Trotter [41]). *Let $\mathbf{P} = (N, P)$ be a poset that is not a linear order. Then, the graph $G_\mathbf{P}$ is bipartite if and only if $\dim(\mathbf{P}) = 2$.*

Using a different approach, Correa and Schulz [10] rediscovered Theorem 3.1 for the vertex cover graph $G^S_\mathbf{P}$ independent of the connection pointed out by Proposition 3.2. Moreover, the following theorem follows easily from Observation 2.1 and Propositions 3.2 and 3.1.

Theorem 3.2. *Problem $1|\text{prec}|\sum w_j C_j$, whenever precedence constraints are given by a k-realizer, has a polynomial-time $(2 - 2/k)$-approximation algorithm.*

A natural question is for which posets one can construct a k-realizer in polynomial time. Yannakakis [45] proved that determining whether the dimension of a poset is at most k is NP-complete for every $k \geq 3$. Moreover, Hegde and Jain [19] recently proved that it is hard to approximate the dimension of a poset with n elements within a factor $n^{0.5-\epsilon}$ in the general case. However, for several special cases including convex bipartite orders (§4.1) and semiorders (§4.2), a minimal realizer can be computed in polynomial time.

Finally, by Proposition 3.1, we remark that $\dim(\mathbf{P})$ and $\chi(G_\mathbf{P})$ are, in general, not the same (see Felsner and Trotter [13] for an example, where $\dim(\mathbf{P})$ is exponentially larger than $\chi(G_\mathbf{P})$). However, it is an immediate consequence of Theorem 3.1 that $\dim(\mathbf{P}) = \chi(G_\mathbf{P})$ when $\dim(\mathbf{P}) = 3$. Therefore, a 3-realizer for a 3-dimensional partial order P (as in §§4.1 and 4.2) immediately gives an optimal coloring for $G_\mathbf{P}$.

4. Precedence constraints with low dimension. In this section, we will apply the previous framework to design approximation algorithms for special cases of posets, namely, convex bipartite orders and semiorders. We note that these results can be generalized to a richer class of posets obtained by the lexicographic sum of posets of the above-mentioned types (as proved for the fractional dimension in §6.4).

4.1. Convex bipartite precedence constraints. In this section, we consider $1|\text{prec}|\sum w_j C_j$ for which the precedence constraints form a so-called convex bipartite order. For this class of partial orders, we show how to construct a realizer of size 3 in polynomial time. By Theorem 3.2, this gives a 4/3-approximation algorithm.

A *convex bipartite order* $\mathbf{P} = (N = J^- \cup J^+, P)$ is defined as follows:

(i) The set of jobs are divided into two disjoint sets $J^- = \{j_1, \ldots j_a\}$ and $J^+ = \{j_{a+1}, \ldots, j_n\}$, known as the *minus jobs* and *plus jobs*, respectively.

(ii) For every $k = a + 1, \ldots, n$, there are two indices $l(k)$ and $r(k)$ with $1 \leq l(k) \leq r(k) \leq a$ such that $(j_i, j_k) \in P$ if and only if $l(k) \leq i \leq r(k)$ (*bipartiteness* and *convexity*).

It is not hard to check that convex bipartite orders can be recognized in polynomial time. Moreover, the class of convex bipartite orders forms a proper subset of the class of general bipartite orders and a proper superset of the class of strong bipartite orders (Möhring [28]). Lemma 4.1 states that the class of convex bipartite orders has dimension ≤ 3 (this result was previously unknown to the best of our knowledge).

Lemma 4.1. *Given a convex bipartite order $\mathbf{P} = (N, P)$, a realizer of size 3 can be computed in polynomial time.*

The proof of Lemma 4.1 can be found in the appendix. Theorem 3.2 and Lemma 4.1 give us the following result.

Theorem 4.1. *Problem $1|\text{prec}|\sum w_j C_j$ for which the precedence constraints form a convex bipartite order has a polynomial-time 4/3-approximation algorithm.*

This result improves upon a previous algorithm by Woeginger [44] that achieved an approximation ratio arbitrarily close to the golden ratio. Furthermore, it is worth noting that $1|\text{prec}|\sum w_j C_j$ with precedence constraints that form a convex bipartite order is not known to be NP-hard.

4.2. Semiorders. A poset $\mathbf{P} = (N, P)$ is a *semiorder* (also called *unit* interval order) if there is a function F that assigns to each $x \in N$ a closed interval $F(x) = [a_x, b_x] \subset \mathbb{R}$ of unit length so that $x < y$ in P if and only if $b_x < a_y$ in $\mathbb{R}$. *Interval orders* are a proper superclass of semiorders that allow arbitrary interval lengths in this definition.

Semiorders can be recognized in $O(n^2)$ time (Möhring [28], Trotter [41]). Moreover, Rabinovitch proved, by constructing a realizer, that the dimension of *semiorders* is at most three (Rabinovitch [33], Trotter [41]). The constructive proof can easily be turned into a polynomial algorithm and, together with Theorem 3.2, we have the following theorem.

Theorem 4.2. *Problem $1|\text{prec}|\sum w_j C_j$ for which the precedence constraints form a semiorder has a polynomial-time 4/3-approximation algorithm.*

Finally, we show that this approach cannot be applied to interval orders. The dimension of interval orders can be of order $\log\log n$ (Trotter [41]). Furthermore, we will show that the graph of incomparable pairs is not colorable with a constant number of colors. To prove this, we use *canonical* interval orders. For an integer $n \geq 2$, let $\mathbf{I}_n$ denote the interval order determined by the set of all closed intervals with distinct integer end points from $[n]$. We will find it convenient to view the elements of $\mathbf{I}_n$ as 2-element subsets of $[n]$ with $(\{i_1, i_2\}, \{i_3, i_4\})$ in $\mathbf{I}_n$ if and only if $i_2 < i_3$ in $\mathbb{R}$ or $\{i_1, i_2\} = \{i_3, i_4\}$. Interval orders of the form $\{\mathbf{I}_N \colon n \geq 2\}$ are called *canonical* (Trotter [42]) (we point out that canonical interval orders are also used to prove that interval orders have unbounded dimension; see, e.g., Trotter [41]).

Theorem 4.3. *For any integer k, there exists an integer n_0 so that, if $n \geq n_0$, then the chromatic number $\chi(G_{\mathbf{I}_n})$ is larger than k.*

Proof. The chromatic number $\chi(G_{\mathbf{I}_n})$ is clearly a nondecreasing function of n. We assume that $\chi(G_{\mathbf{I}_n}) \leq k$ for all $n \geq 2$, and we obtain a contradiction when n is sufficiently large.

Let the map $\varphi \colon \binom{[n]}{3} \to \{1, 2, \ldots, k\}$ denote a coloring of the 3-element subsets of $[n]$. Note that any coloring of $G_{\mathbf{I}_n}$ defines the map φ by letting $\varphi(\{i, j, l\})$ equal the coloring of the vertex $(\{i, j\}, \{j, l\})$[3] in $G_{\mathbf{I}_n}$.

Let n_0 equal the Ramsey number $R(3\colon h_1, h_2, h_3, \ldots, h_k)$, where $h_1 = h_2 = \cdots = h_k = 4$. Now, pick n to be greater than or equal to n_0 and hence $|[n]| \geq n_0$. Consider any coloring of $G_{\mathbf{I}_n}$ and the corresponding map φ. By Ramsey's theorem (Trotter [41]), there exists a subset H of $[n]$ with $|H| \geq 4$ so that $\varphi(A) = c$ for every 3-element subset A of H. Consider $\{i, j, l, m\} \subseteq H$, where $i < j < l < m$. We know that $\varphi(\{i, j, l\}) = c$ and $\varphi(\{j, l, m\}) = c$. However, this implies that the adjacent vertices $(\{i, j\}, \{j, l\})$, and $(\{j, l\}, \{l, m\})$ are colored with the same color. The vertices are adjacent because $\{(\{j, l\}, \{i, j\}), (\{l, m\}, \{j, m\})\}$ forms an alternating cycle.

Thus, for *any* k-coloring, we have two adjacent nodes in $G_{\mathbf{I}_n}$, which are colored by the same color. This contradicts the existence of a valid k-coloring for $G_{\mathbf{I}_n}$ when $n \geq n_0$. $\square$

5. Scheduling with low fractional dimension. In this section, we observe that better than 2-approximation algorithms are possible provided that the set of precedence constraints has low fractional dimension. Applications that follow this pattern are given in §6.

We say that a poset $\mathbf{P}$ admits an *efficiently samplable* $k\colon t$-realizer if there exists a randomized algorithm that, in time polynomial in the size of the poset, returns any linear extension from a k-fold realizer $\mathcal{F} = \{L_1, L_2, \ldots, L_t\}$ with probability $1/t$.

The following theorem generalizes Theorem 3.2. Its proof uses an argument similar to that of the Hochbaum [20] original proof of Theorem 3.2 and yields an approximation result based on the fractional coloring number of the vertex cover graph.

Theorem 5.1. *The problem $1|\text{prec}|\sum w_j C_j$, whenever precedence constraints admit an efficiently samplable $k\colon t$-realizer, has a randomized $(2 - 2/(t/k))$-approximation algorithm.*

[3] Note that we can assume without loss of generality that $i < j < l$.

Proof. Let S be an instance of $1|\text{prec}|\sum w_j C_j$, where precedence constraints are given by a poset $\mathbf{P} = (N, P)$ that admits an efficiently samplable $k{:}t$-realizer $\mathcal{F} = \{L_1, L_2, \ldots, L_t\}$. Furthermore, we assume that $\text{fdim}(\mathbf{P}) \geq 2$. The case when $\text{fdim}(\mathbf{P}) = 1$, i.e., when $\mathbf{P}$ is a linear order, is trivial.

Let $G_{\mathbf{P}}^S = (V_{\mathbf{P}}, E_{\mathbf{P}})$ be the weighted vertex cover instance associated with S, where each vertex (incomparable pair) $(i, j) \in V_{\mathbf{P}}$ has weight $w_{(i,j)} = p_i \cdot w_j$ as specified in §2.2. Using a result by Nemhauser and Trotter [29], the [CS-LP] formulation of $G_{\mathbf{P}}^S$ can be solved combinatorially and in polynomial time, and has half integral extreme point solutions. Furthermore, let V_i be the set of vertices with value i $(i = 0, \frac{1}{2}, 1)$ in the optimum solution. Denote by $G_{\mathbf{P}}^S[V_{1/2}]$ the subgraph of $G_{\mathbf{P}}^S$ induced by the vertex set $V_{1/2}$. We consider the linear extensions of $\mathcal{F}$ as outcomes in a uniform sample space. For an incomparable pair (x, y), the *probability that y is over x in $\mathcal{F}$* is given by

$$\text{Prob}_{\mathcal{F}}[y > x] = \frac{1}{t}\big|\{i = 1, \ldots, T\colon\, y > x \in L_i\}\big| \geq \frac{k}{t}. \tag{4}$$

The last inequality holds because every incomparable pair is reversed in at least k linear extensions of $\mathcal{F}$.

Let us pick one linear extension L uniformly at random from $\mathcal{F} = \{L_1, \ldots, L_t\}$. Then, by linearity of expectation, the expected value of the independent set $I_{1/2}$, obtained by taking the incomparable pairs in $V_{1/2}$ that are reversed in L, is[4]

$$\text{E}[w(I_{1/2})] = \sum_{(i,j)\in V_{1/2}} \text{Prob}_{\mathcal{F}}[j > i] \cdot w_{(i,j)} \geq \frac{k}{t} \cdot w(V_{1/2}). \tag{5}$$

A vertex cover solution C for the graph $G_{\mathbf{P}}^S[V_{1/2}]$ can be obtained by picking the nodes that are not in $I_{1/2}$, namely, $C = V_{1/2} \backslash I_{1/2}$. The expected value of this solution is

$$\text{E}[w(C)] = w(V_{1/2}) - \text{E}[w(I_{1/2})] \leq \left(1 - \frac{k}{t}\right) w(V_{1/2}).$$

As observed in Hochbaum [20], $V_1 \cup C$ gives a valid vertex cover for the graph $G_{\mathbf{P}}^S$. Moreover, the expected value of the cover is bounded as follows:

$$\begin{aligned}
\text{E}[w(V_1 \cup C)] &\leq w(V_1) + \left(1 - \frac{k}{t}\right) w(V_{1/2}) && (6)\\
&\leq 2\left(1 - \frac{k}{t}\right)\left(w(V_1) + \frac{1}{2} w(V_{1/2})\right) && (7)\\
&\leq \left(2 - \frac{2}{t/k}\right) \text{OPT}, && (8)
\end{aligned}$$

where the last inequality holds because $w(V_1) + \frac{1}{2} w(V_{1/2})$ is the optimal value of [CS-LP]. Note that $t/k \geq \text{fdim}(\mathbf{P}) \geq 2$ was used for the second inequality. Theorem 2.1 implies that any α-approximation algorithm for $G_{\mathbf{P}}^S$ also gives an α-approximation algorithm for S. Thus, we obtain a randomized $(2 - 2/(t/k))$-approximation algorithm for S. □

6. Precedence constraints with low fractional dimension.

6.1. Interval orders.

A poset $\mathbf{P} = (N, P)$ is an *interval order* if there is a function F that assigns to each $x \in N$ a closed interval $F(x) = [a_x, b_x]$ of the real line $\mathbb{R}$, so that $x < y$ in P if and only if $b_x < a_y$ in $\mathbb{R}$. Interval orders can be recognized in polynomial time, e.g., in $O(n^2)$-time using Papadimitriou and Yannakakis [30]. The dimension of interval orders can be of order $\log\log n$ (Trotter [41]) whereas the fractional dimension is known to be less than four (Brightwell and Scheinerman [7]), and this bound is asymptotically tight (Felsner and Trotter [12]). In the following, we show how to obtain a 3/2-approximation algorithm for $1|\text{prec}|\sum w_j C_j$ when the precedence constraints form an interval order. By Theorem 5.1, it is sufficient to prove that interval orders admit an efficiently samplable $k{:}t$-realizer with $t/k = 4$.

Given a poset $\mathbf{P} = (N, P)$, disjoint subsets A and B of the ground set N, and a linear extension L of P, we say that *B is over A in L* if, for every incomparable pair of elements (a, b) with $a \in A$ and $b \in B$, one has $b > a$ in L. The following property of interval orders is fundamental for our approach.

[4] That $I_{1/2}$ is an independent set is easily seen from the fact that a linear ordering L naturally defines a vertex cover by the incomparable pairs $(a, b) \in L$. The incomparable pairs that are reversed in L, namely, $(b, a) \notin L$, form the complement of this set.

THEOREM 6.1 (RABINOVITCH [33]). *A poset $\mathbf{P} = (N, P)$ is an interval order if and only if, for every pair (A, B) of disjoint subsets of N, there is a linear extension L of P with B over A.*

By using this property, we can easily obtain a k-fold realizer $\mathcal{F} = \{L_1, \dots, L_t\}$ with $k = 2^{n-2}$ and $t = 2^n$, where $n = |N|$. Indeed, consider every subset A of N and let L_A be a linear extension of P in which $B = N \backslash A$ is over A. Now, let $\mathcal{F}$ be the multiset of all the L_As. Note that $|\mathcal{F}| = 2^n$. Moreover, for any incomparable pair (x, y), there are at least $k = 2^{n-2}$ linear extensions in $\mathcal{F}$ for which $x \in B$ and $y \in A$. Finally, observe that we can efficiently pick uniformly at random one linear extension from $\mathcal{F}$ for every job $j \in N$, put j either in A or in B with the same probability $1/2$.

By the previous observations and Theorem 5.1, we have a randomized polynomial-time $3/2$-approximation for $1 \mid \text{prec} \mid \sum w_j C_j$ when the precedence constraints form an interval order. The algorithm described here can easily be derandomized by using the standard method of conditional probabilities.

THEOREM 6.2. *Problem $1 \mid \text{prec} \mid \sum w_j C_j$ with precedence constraints forming an interval order has a deterministic polynomial-time $3/2$-approximation algorithm.*

6.2. Interval dimension 2. The *interval dimension* of a poset $\mathbf{P} = (N, P)$, denoted by $\dim_I(\mathbf{P})$, is defined (Trotter [41]) as the least t for which there exist t extensions $Q_1, Q_2, \dots, Q_t$, so that:
- $P = Q_1 \cap Q_2 \cap \cdots \cap Q_t$, and
- (N, Q_i) is an interval order for $i = 1, 2, \dots, t$.

Obviously, if $\mathbf{P}$ is an interval order, then $\dim_I(\mathbf{P}) = 1$. In general, because a linear order is an interval order, $\dim_I(\mathbf{P}) \le \dim(\mathbf{P})$.

The class of posets of interval dimension 2 forms a proper superclass of the class of interval orders. Posets of interval dimension 2 can be recognized in $O(n^2)$ time per Ma and Spinrad [26]. Given a poset $\mathbf{P}$ with $\dim_I(\mathbf{P}) = 2$, their algorithm also yields an interval realizer $\{Q_1, Q_2\}$. As described in §6.1, we obtain k-fold realizers $\mathcal{F}_1 = \{L_1, L_2, \dots, L_t\}$ and $\mathcal{F}_2 = \{L'_1, L'_2, \dots, L'_t\}$ of Q_1 and Q_2, respectively, with $k = 2^{n-2}$ and $t = 2^n$. It is immediate that $\mathcal{F} = \mathcal{F}_1 \cup \mathcal{F}_2$ is a k-fold realizer of $\mathbf{P}$ of size $2t = 2^{n+1}$. Furthermore, we can efficiently pick uniformly at random one linear extension from $\mathcal{F}$: Pick uniformly at random a linear extension from either $\mathcal{F}_1$ or $\mathcal{F}_2$ with the same probability $1/2$. Again, by using conditional probabilities, we have the following.

THEOREM 6.3. *Problem $1 \mid \text{prec} \mid \sum w_j C_j$, whenever precedence constraints have interval dimension at most 2, has a polynomial-time 1.75-approximation algorithm.*

6.3. Posets of bounded degree. In the following, we will see how to use Theorem 5.1 to obtain approximation algorithms for $1 \mid \text{prec} \mid \sum w_j C_j$ when the precedence constraints form a poset of bounded degree. Before we proceed, we need to introduce some definitions.

Let $\mathbf{P} = (N, P)$ be a poset. For any job $j \in N$, define the *degree of* j, denoted $\deg(j)$, as the number of jobs comparable (but not equal) to j in $\mathbf{P}$. Given a job j, let $D(j)$ denote the set of all jobs that are less than j and $U(j)$ those that are greater than j in P. Define $\deg_D(j) = |D(j)|$ and $\Delta_D(\mathbf{P}) = \max\{\deg_D(j)\colon j \in N\}$. The quantities $\deg_U(j)$ and $\Delta_U(\mathbf{P})$ are defined analogously.

The NP-completeness proof for $1 \mid \text{prec} \mid \sum w_j C_j$ given by Lenstra and Rinnoy Kan [25] was actually provided for posets $\mathbf{P}$ with $\Delta_D(\mathbf{P}) = 2$. By using fractional dimension, we show that these posets (with bounded $\min\{\Delta_D, \Delta_U\}$) allow for better than 2-approximation.

THEOREM 6.4. *Problem $1 \mid \text{prec} \mid \sum w_j C_j$ has a polynomial-time $(2 - 2/f)$-approximation algorithm, where $f = 1 + \max\{\min\{\Delta_D, \Delta_U\}, 1\}$.*

PROOF. Let $\mathbf{P} = (N, P)$ be the poset representing the precedence constraints with bounded $\min\{\Delta_D, \Delta_U\}$. Assume, without loss of generality, that $\mathbf{P}$ is *not* decomposable with respect to lexicographic sums (see §6.4).[5] Otherwise, a decomposition with respect to lexicographic sums can be done in $O(n^2)$ time (see, e.g., Möhring [28]) and each component can be considered separately.

For any permutation M of N, consider the set $C(M)$ of critical pairs (x, y) that satisfy the following two conditions:

(i) $x > (D(y) \cup \{y\})$ in M if $|D(y)| < \Delta_D$, and
(ii) $x > D(y)$ in M if $|D(y)| = \Delta_D$.

[5] This is needed in order to be able to apply the algorithm of Felsner and Trotter [12] next.

Felsner and Trotter [12] present an algorithm that converts in polynomial time a permutation M of N to a linear extension L of P so that L reverses all critical pairs in the set $C(M)$. Now, set $t=|N|!$ and consider the set $\mathcal{M}=\{M_1, M_2, \ldots, M_t\}$ of all permutations of the ground set N. Observe that, for any critical pair (x, y), there are at least $n!/(\Delta_D+1)$ different permutations $M_i \in \mathcal{M}$ where the critical pair is reversed, i.e., $(y, x)\in C(M_i)$. This is because any ordering of the n elements defines an ordering of $T:=\{x\}\cup D(y)$ (respectively, an ordering of $T:=\{x\}\cup\{y\}\cup D(y)$) and, in $1/|T|\geq 1/(\Delta_D+1)$ of them, x is the last element of T. Applying the algorithm in Felsner and Trotter [12], we obtain a k-fold realizer $\mathcal{F}=\{L_1, \ldots, L_t\}$ of P with $t=n!$ and $k=n!/(\Delta_D+1)$. Moreover, we can efficiently pick uniformly at random one linear extension from $\mathcal{F}$: Generate uniformly at random one permutation of jobs (e.g., by using Knuth's shuffle algorithm) and transform it into a linear extension with the described properties by using the algorithm in Felsner and Trotter [12]. The described algorithm can be derandomized by using the classical method of conditional probabilities. Finally, observe that we can repeat a similar analysis by using Δ_U instead of Δ_D. □

We remark that it is necessary to use the *fractional* dimension for obtaining the above-mentioned result. To see this, consider the *incidence* poset $\mathbf{P}(G)=(N, P)$ defined as follows: Given an undirected graph $G(V, E)$, let $N=V\cup E$ and, for every $v\in V$ and $e=\{v_1, v_2\}\in E$, put $(v, e)\in P$ if and only if $v\in\{v_1, v_2\}$. Because every edge is adjacent to only two vertices, Δ_D is bounded by 2. For the complete graph on n nodes, K_n, Spencer [38] showed that $\dim(\mathbf{P}(K_n))=\Theta(\log\log n)$. From the discussion here, however, we have $\operatorname{fdim}(\mathbf{P}(K_n))\leq 1+\min\{\Delta_U, \Delta_D\}\leq 3$.

6.4. Lexicographic sums. In this section, we show how to use previous results to obtain approximation algorithms for new ordered sets. The construction we use, *lexicographic sums*, comes from a very simple pictorial idea (see Trotter [41] for a comprehensive discussion). Take a poset $\mathbf{P}=(N, P)$ and replace each of its points $x\in N$ with a partially ordered set $\mathbf{Q_x}$ (the *module*) such that the points in the module have the same relation to points outside it. A more formal definition follows. For a poset $\mathbf{P}=(N, P)$ and a family of posets $\mathcal{S}=\{(Y_x, Q_x)\mid x\in N\}$ indexed by the elements in N, the lexicographic sum of $\mathcal{S}$ over (N, P) (denoted $\sum_{x\in(N, P)}(Y_x, Q_x)$) is the poset (Z, R), where $Z=\{(x, y)\mid x\in N,\ y\in Y_x\}$ and $(x_1, y_1)\leq(x_2, y_2)$ in R if and only if one of the following two statements holds:

(i) $x_1<x_2$ in P, and
(ii) $x_1=x_2$ and $y_1\leq y_2$ in Q_{x_1}.

We call $\mathcal{P}=P\cup\mathcal{S}$ the *components* of the lexicographic sum. A lexicographic sum is *trivial* if $|N|=1$ or if $|Y_x|=1$ for all $x\in N$. A poset is *decomposable with respect to lexicographic sums* if it is isomorphic to a nontrivial lexicographic sum.

In case the precedence constraints of every component admit an efficiently samplable realizer, we observe that this translates into a randomized approximation algorithm.

THEOREM 6.5. *Problem* $1\mid\text{prec}\mid\sum w_jC_j$, *whenever precedence constraints form a lexicographic sum whose components* $i\in\mathcal{P}$ *admit efficiently samplable realizers, has a polynomial-time randomized* $(2-2/(t/k))$-*approximation algorithm, where* $t/k=\max_{i\in\mathcal{P}}(t_i/k_i)$.

Finally, we point out that, if the approximation algorithm for each component can be derandomized, this yields a derandomized approximation algorithm for the lexicographic sum. In particular, this can be done when all components have low dimension.

7. NP-completeness for interval orders. In this section, we show that $1\mid\text{prec}\mid\sum w_jC_j$ remains NP-hard even in the special case of interval order precedence constraints.

To prove this, we exploit the vertex cover nature of problem $1\mid\text{prec}\mid\sum w_jC_j$. Finding an optimum solution to a scheduling instance S, where precedence constraints are given by an interval order $\mathbf{I}$, is equivalent to solving the minimum weighted vertex cover problem in the graph $G_{\mathbf{I}}^S$ (see §2.2).

THEOREM 7.1. *Problem* $1\mid\text{prec}\mid\sum w_jC_j$ *with precedence constraints that form an interval order is* NP-*hard.*

PROOF. A graph G is said to have bounded degree d if every vertex v in G is adjacent to at most d other vertices. The problem of deciding if a graph G with bounded degree 3 has a (unweighted) vertex cover of size at most m is NP-complete (Garey et al. [14]). We provide a reduction from the minimum vertex cover problem on graphs with bounded degree 3 to $1\mid\text{prec}\mid\sum w_jC_j$ with interval order precedence constraints.

Given a connected graph $G=(V, E)$ with bounded degree 3, we construct an instance S of $1\mid\text{prec}\mid\sum w_jC_j$ with interval order precedence constraints so that the graph $G_{\mathbf{I}}^S$ has a weighted vertex cover with value less than

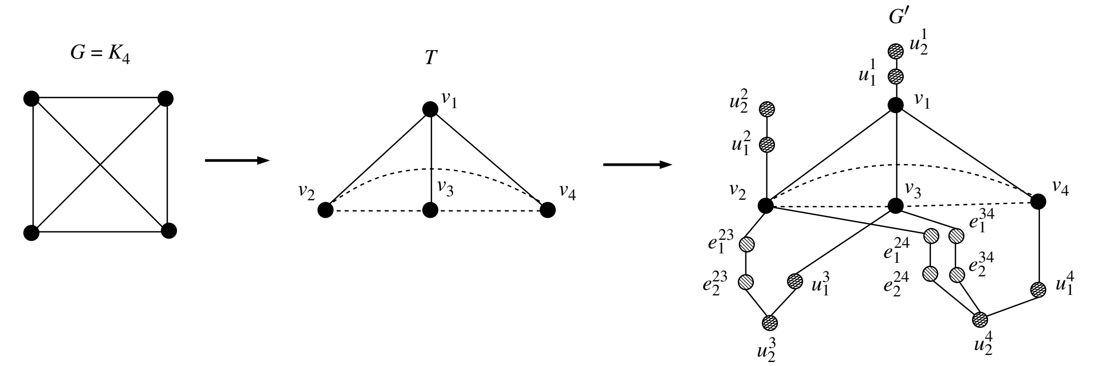

FIGURE 1. The breadth-first search tree $T = (V, E_T)$ for the graph $G = K_4$ and the graph G' are shown. In the drawing of T, the solid edges belong to E_T.

$m + c + 1$ if and only if G has a vertex cover of size at most m, where c is a fixed value defined later (see Equation (9)). We present the construction of S in two stages.

Stage 1 (*Tree layout of the graph*). Starting from an arbitrary but fixed vertex $s \in V$, consider the tree $T = (V, E_T)$, with $E_T \subseteq E$, rooted at s on the set of nodes reachable from s by using, for example, breadth-first search. Furthermore, we number the vertices of T from the top down and from left to right. Figure 1 shows the breadth-first search tree T for K_4.

Define $G' = (V', E')$ to be the graph obtained from T in the following way. For each vertex v_i in T, we add two new vertices u_2^i and u_1^i and edges $\{u_2^i, u_1^i\}$ and $\{u_1^i, v_i\}$. Furthermore, for each edge $\{v_i, v_j\} \in E \backslash E_T$ with $i < j$, we add vertices e_1^{ij} and e_2^{ij} and edges $\{v_i, e_1^{ij}\}$, $\{e_1^{ij}, e_2^{ij}\}$, and $\{e_2^{ij}, u_2^j\}$.

The following claim relates optimum unweighted vertex covers of G and G'. Its proof is similar to the proof in Alimonti and Kann [1] for proving APX-completeness of vertex cover on cubic graphs.

CLAIM 1. Let $C_* \subseteq V$ and $C'_* \subseteq V'$ be optimum vertex cover solutions to G and G', respectively. Then, $|C_*| = |C'_*| - |V| - |E \backslash E_T|$.

PROOF OF CLAIM 1. It is easy to see that, from every vertex cover $C \subseteq V$ of G, we can construct a vertex cover $C' \subseteq V'$ of G' of size exactly $|C| + |V| + |E \backslash E_T|$. In C', we include u_1^i for all i with $v_i \in V \backslash C$, u_2^i for all i with $v_i \in C$, e_1^{ij} for each $\{v_i, v_j\} \in E \backslash E_T$ with $v_i \in V \backslash C$, e_2^{ij} for each $(v_i, v_j) \in E \backslash E_T$ with $v_i \in C$, and every vertex in C.

Given a vertex cover $C' \subseteq V'$ of G', we transform it into a vertex cover $C \subseteq V$ of G in the following manner. Suppose there exist $v_i, v_j \in V$ with $i < j$ such that $\{v_i, v_j\} \in E$ and $v_i \notin C'$, $v_j \notin C'$. Because C' is a feasible vertex cover of G', we have that $\{v_i, v_j\} \in E \backslash E_T$ and either $\{e_1^{ij}, e_2^{ij}, u_1^j\} \subseteq C'$ or $\{e_1^{ij}, u_2^j, u_1^j\} \subseteq C'$. Thus, we can obtain a vertex cover $C'' \subseteq V'$ of G' with $|C''| \le |C'|$ by letting $C'' = (C' \backslash \{u_1^j, e_2^{ij}\}) \cup \{v_j, u_2^j\}$. Repeating this procedure will result in a vertex cover $C''' \subseteq V'$ of G' with $|C'''| \le |C'|$ such that $C = C''' \cap V$ is a feasible vertex cover of G. Furthermore, it is easy to see that $|C| \le |C'''| - |V| - |E \backslash E_T|$. $\square$

Stage 2 (*Construction of scheduling instance*). Given the vertex cover graph $G = (V, E)$ and its corresponding tree $T = (V, E_T)$, we construct the scheduling instance S with processing times, weights, and precedence constraints to form an interval order $\mathbf{I}$ as defined next (see Figure 2 for an example), where k is a large value to be determined later.

Job	Interval repr.	Proc. time	Weight
s_0	$[-1, 0]$	1	0
s_1	$[0, 1]$	$1/k$	1
s_j, $j = 2, \ldots, \lvert V \rvert$	$[i, j]$, where $\{v_i, v_j\} \in E_T$, $i < j$	$1/k^j$	k^i
m_i, $i = 1, \ldots, \lvert V \rvert$	$[i - \frac{1}{2}, \lvert V \rvert + i]$	$1/k^{(\lvert V \rvert + i)}$	k^i
e_i, $i = 1, \ldots, \lvert V \rvert$	$[\lvert V \rvert + i, \lvert V \rvert + i + 1]$	0	$k^{(\lvert V \rvert + i)}$
b_{ij}, where $\{v_i, v_j\} \in E \backslash E_T$, $i < j$	$[i, j - \frac{1}{2}]$	$1/k^j$	k^i

REMARK 7.1. Let i and j be two jobs in S with interval representations $[a, b]$ and $[c, d]$, respectively, and where $a \le d$. By the construction of the scheduling instance S, we have $p_i \le 1/k^{\lceil b \rceil}$ and $w_j \le k^{\lceil c \rceil}$. It follows that $p_i \cdot w_j = 1$ or $p_i \cdot w_j \le 1/k$ if i and j are incomparable because $p_i \cdot w_j \ge k$ implies that $b < c$, i.e., i's interval representation is completely to the left of j's interval representation. Furthermore, if $p_i \cdot w_j = 1$, then $\lceil b \rceil = \lceil c \rceil$.

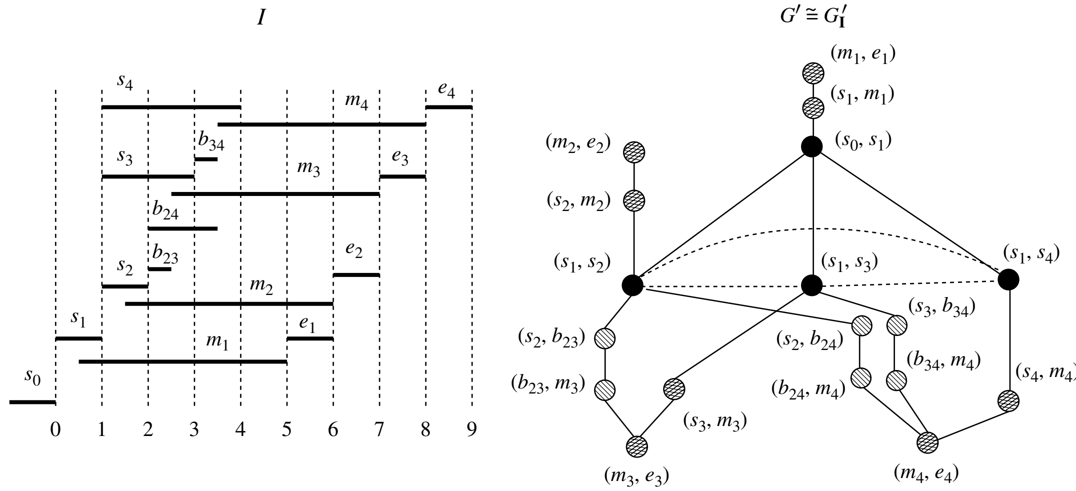

FIGURE 2. The interval order $\mathbf{I}$ obtained from K_4. $G'_{\mathbf{I}}$ is the subgraph induced on the graph $G^S_{\mathbf{I}}$ by the vertex subset D (the vertices with weight 1).

Let

$$\begin{aligned} D = &\{(s_0, s_1)\} \\ &\cup \big\{(s_i, s_j)\colon v_i \text{ is the parent of } v_j \text{ in } T\big\} \\ &\cup \big\{(s_i, m_i), (m_i, e_i)\colon i = 1, 2, \ldots, |V|\big\} \\ &\cup \big\{(s_i, b_{ij}), (b_{ij}, m_j)\colon \{v_i, v_j\} \in E \backslash E_T,\ i < j\big\}. \end{aligned}$$

By the interval representation of the jobs and Remark 7.1, we have the following.

CLAIM 2. A pair of incomparable jobs (i, j) has $p_i \cdot w_j = 1$ if $(i, j) \in D$. Otherwise, if $(i, j) \notin D$, then $p_i \cdot w_j \leq 1/k$.

CLAIM 3. Let $G'_{\mathbf{I}} = (D, E_{\mathbf{I}})$ be the subgraph of $G^S_{\mathbf{I}}$ induced by the vertex subset D. Then, $G'_{\mathbf{I}}$ and G' are isomorphic.

PROOF OF CLAIM 3. We relate the two graphs $G'_{\mathbf{I}}$ and G' by the bijection $f\colon D \to V'$, defined as follows:

$$f((a, b)) = \begin{cases} v_j & \text{if } (a, b) = (s_i, s_j), \\ u^i_1 & \text{if } (a, b) = (s_i, m_i), \\ u^i_2 & \text{if } (a, b) = (m_i, e_i), \\ e^{ij}_1 & \text{if } (a, b) = (s_i, b_{ij}), \\ e^{ij}_2 & \text{if } (a, b) = (b_{ij}, m_j). \end{cases}$$

Suppose $\{(a, b), (c, d)\} \in E_{\mathbf{I}}$. By the fact that I is an interval order (i.e., does not contain any $\mathbf{2} + \mathbf{2}$ structures; see Figure 3), as induced posets (Trotter [41]), together with the definition of D, we can assume that $b = c$ and $a \neq d$.

Now, consider the possible cases of $\{(a, b), (b, d)\}$.

$(a = s_i,\ b = s_j,\ d = s_k,\ i < j < k)$ By construction of I, v_j is the parent of v_k, i.e., $\{f((s_i, s_j)), f((s_j, s_k))\} = \{v_j, v_k\} \in E_T \subseteq E'$.

$(a = s_i,\ b = s_j,\ d = b_{jk},\ i < j < k)$ Then, $f((s_i, s_j)) = v_j$ and $f((s_j, b_{jk})) = e^{jk}_1$ and, by definition of G', we have $\{v_j, e^{jk}_1\} \in E'$.

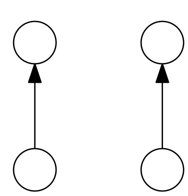

FIGURE 3. A $\mathbf{2} + \mathbf{2}$ poset.

The remaining cases $(a = s_i,\ b = s_j,\ d = m_j,\ i < j)$, $(a = s_i,\ b = b_{ij},\ d = m_j,\ i < j)$, $(a = s_i,\ b = m_i,\ d = e_i)$, and $(a = b_{ij},\ b = m_j,\ d = e_j,\ i < j)$ are similar to these and it is straightforward to check the implication $\{(a, b), (b, d)\} \in E_{\mathbf{I}} \Rightarrow \{f((a, b)), f((b, c))\} \in E'$.

On the other hand, suppose $\{a, b\} \in E'$ and again consider the different possible cases.

$(a = v_i,\ b = v_j,\ i < j)$ Then, v_i is the parent of v_j in T and $f^{-1}(v_i) = (s_k, s_i)$ and $f^{-1}(v_j) = (s_i, s_j)$ for some $k < i < j$. Because the interval representation of s_k is completely to the left of the interval representation of s_j in I, the incomparable pairs (s_k, s_i) and (s_i, s_j) cannot be reversed in the same linear extension, i.e., $\{(s_k, s_i), (s_i, s_j)\} \in E_{\mathbf{I}}$.

$(a = v_i,\ b = e_1^{ij},\ i < j)$ Then, $f^{-1}(v_i) = (s_k, s_i)$ and $f^{-1}(e_1^{ij}) = (s_i, b_{ij})$ for some $k < i < j$. Because the interval representation of s_k is completely to the left the interval representation of b_{ij} in I, the incomparable pairs (s_k, s_i) and (s_i, b_{ij}) cannot be reversed in the same linear extension, i.e., $\{(s_k, s_i), (s_i, b_{ij})\} \in E_{\mathbf{I}}$.

The remaining cases $(a = e_1^{ij}, b = e_2^{ij}, i < j)$, $(a = e_2^{ij}, b = u_2^j, i < j)$, $(a = u_1^j, b = u_2^j)$, and $(a = v_j, b = u_1^j)$ are similar to these and are omitted.

We have thus proved that $\{(a, b), (b, d)\} \in E_{\mathbf{I}} \Leftrightarrow \{f((a, b)), f((b, c))\} \in E'$, i.e., the function f defines an isomorphism between $G'_{\mathbf{I}}$ and G'. □

By Claim 2, each incomparable pair of jobs $(i, j) \notin D$ satisfies $p(i) \cdot w(j) \leq 1/k$. Let n be the number of jobs in the scheduling instance S and select k to be $n^2 + 1$. Let $C_{\mathbf{I}}$ and $C'_{\mathbf{I}}$ be minimum vertex covers of $G^S_{\mathbf{I}}$ and $G'_{\mathbf{I}}$ and denote their respective values by $w(C_{\mathbf{I}})$ and $w(C'_{\mathbf{I}})$. Because $G'_{\mathbf{I}}$ is unweighted, we have $w(C'_{\mathbf{I}}) = |C'_{\mathbf{I}}|$. By the selection of k and Claim 2, we have

$$\sum_{(i,j)\in \mathrm{inc}(I)\setminus D} p_i w_j < 1 \quad \text{and thus } w(C'_{\mathbf{I}}) = \lfloor w(C_{\mathbf{I}}) \rfloor.$$

Because $G'_{\mathbf{I}}$ and G' are isomorphic (Claim 3), we have by Claim 1 that graph G has an optimal vertex cover less at most m if and only if $\lfloor w(C_{\mathbf{I}}) \rfloor \leq m + c$, where

$$c = |V| + |E \setminus E_T|. \quad \square \tag{9}$$

We remark that this proof implies that it is NP-hard to approximate the minimum weighted vertex cover problem within some factor $r > 1$ on the family of graphs $G^S_{\mathbf{I}}$ obtained from scheduling instances with interval order precedence constraints.

8. Hardness of variable cost. We show that approximating the variable cost of scheduling problem $1 \mid \mathrm{prec} \mid \sum w_j C_j$ is equivalent to approximating the vertex cover problem. Theorem 2.1 implies that minimizing the variable cost of $1 \mid \mathrm{prec} \mid \sum w_j C_j$ is a special case of vertex cover and therefore is not harder to approximate. It remains to prove the other direction. We do so by proving that, for any graph G, we can construct a scheduling instance for which minimizing the variable cost is essentially equal to finding a minimum vertex cover of G.

Theorem 8.1. *Approximating the variable cost of* $1 \mid \mathrm{prec} \mid \sum w_j C_j$ *is as hard as approximating vertex cover.*

Proof. Let $G = (V, E)$ be a vertex cover instance and let $n = |V|$. We will construct a scheduling instance S as follows. The construction is inspired by the so-called *adjacency* poset of G. Let $r \geq 1$, $\epsilon > 0$ and choose $k > n^2 r/\epsilon$. For each vertex $v_i \in V$, there are two jobs v'_i and v''_i. The processing time and weight for a job v'_i are $1/k^i$ and 0, respectively. Conversely, the processing time and weight for a job v''_i are 0 and k^i, respectively.

S has the following precedence constraints: For each edge $\{v_i, v_j\} \in E$, the precedence constraints $v'_i < v''_j$ and $v'_j < v''_i$. Finally, we add $v'_i < v''_j$ for every i, j with $i < j$. See Figure 4 for a small example.

Now, consider the graph $G^S_{\mathbf{P}}$. It has at most n^2 vertices. The n vertices corresponding to the incomparable pairs (v'_i, v''_i) have weight 1. All other vertices have weight at most $1/k$, which, by the choice of k, is very small. The total weight of these "light" vertices is no more than n^2/k.

Moreover, the subgraph induced by the vertices with weight 1 is isomorphic to G. To see this, recall that there is an edge between the vertices (v'_i, v''_i) and (v'_j, v''_j) in $G^S_{\mathbf{P}}$ if and only if both precedence constraints $v'_i \to v''_j$ and $v'_j \to v''_i$ are present in S. This, in turn, is the case if and only if $\{v_i, v_j\} \in E$.

Using the connection between S and $G^S_{\mathbf{P}}$ provided by Theorem 2.1 and the close relation between $G^S_{\mathbf{P}}$ and G, it is easy to see that an r-approximation algorithm for the variable cost of $1 \mid \mathrm{prec} \mid \sum w_j C_j$ would imply an approximation algorithm for vertex cover with approximation ratio $r(1 + n^2/k) < (r + \epsilon)$. □

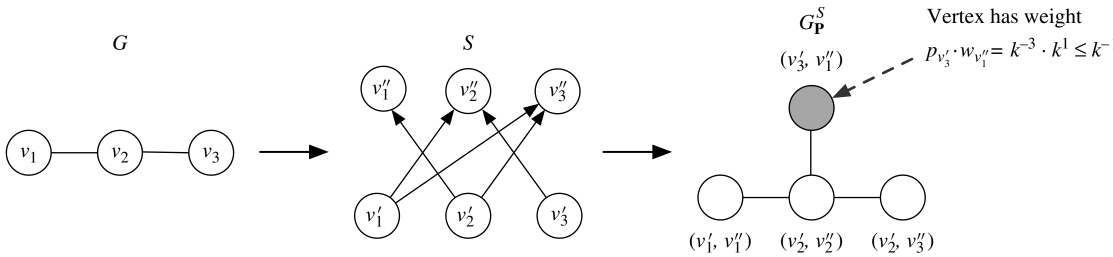

FIGURE 4. The transformation of a graph G.

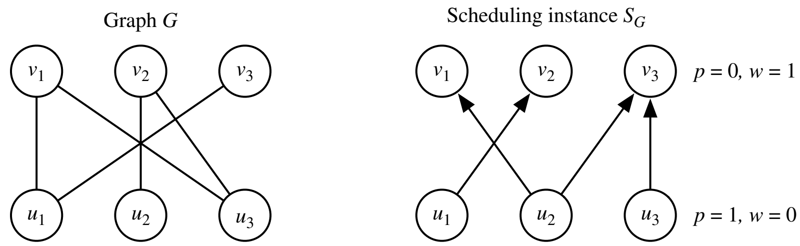

FIGURE 5. An example of a graph G with its associated scheduling instance S_G.

We point out that this reduction fails to yield inapproximability results if the complete objective function (i.e., the fixed cost plus the variable cost) is considered: The fixed cost introduced during the reduction dominates the objective function value, which makes any feasible solution close to optimal. Nevertheless, one can rule out, under some fairly standard assumption, the existence of a PTAS for $1|\text{prec}|\sum w_j C_j$ by establishing a connection between the maximum edge biclique problem and $1|\text{prec}|\sum w_j C_j$. This is done in the next section.

9. Ruling out a PTAS. We uncover a nice relationship between $1|\text{prec}|\sum w_j C_j$ and the maximum edge biclique problem, defined next. This relationship together with an inapproximability result for maximum edge biclique (MEB) (Ambühl et al. [4]) yields Theorem 9.2, i.e., that the scheduling problem has no PTAS unless there is a probabilistic algorithm with running time 2^{N^ϵ} that decides whether a given instance of the satisfiability problem (SAT) is satisfiable, where N is the instance size and $\epsilon > 0$ can be made arbitrarily close to 0.

DEFINITION 9.1. Given an n-by-n bipartite graph G, the *maximum edge biclique problem* is to find a k_1-by-k_2 complete subgraph of G that maximizes $k_1 \cdot k_2$.

With an n-by-n bipartite graph $G = (U, V, E)$, we associate a bipartite scheduling instance S_G with jobs $U \cup V$ and precedence constraints $P = (U \times V) \backslash E$. The jobs of U have processing time 1 and weight 0, and the jobs of V have processing time 0 and weight 1. See Figure 5 for a small example.

The intuition behind the relationship between $1|\text{prec}|\sum w_j C_j$ and MEB is best seen by considering two-dimensional Gantt charts, first introduced by Eastman et al. [11] and later revived by Goemans and Williamson [15] to give elegant proofs for various results related to $1|\text{prec}|\sum w_j C_j$. In a two-dimensional Gantt chart, we have a horizontal axis of processing time and a vertical axis of weight. For a scheduling instance of the above-mentioned form, the chart starts at point $(0, n)$ and ends at point $(n, 0)$. A job j is represented by a rectangle of length p_j and height w_j. Hence, a job of U is represented by a horizontal line of length 1, and a job of V is represented by a vertical line of length 1. Any schedule (linear extension of the jobs) is represented in the two-dimensional Gantt chart by placing the corresponding rectangles of the jobs in the order of the schedule such that the start point of a job is the end point of the previous job (or $(0, n)$ for the first job). The value $\sum_j w_j C_j$ of a schedule is then the area under the "work line" (see the shaded area in Figure 6) or, equivalently, the area above the work line subtracted from n^2. The relationship to MEB now becomes clear from the following observation: Each starting point (s, t) of a job on the work line of a schedule of S_G defines an edge biclique of G of size $(n-s)(n-t)$ by taking the vertices corresponding to the jobs of U that complete after s (there are $n-s$ of them) and the jobs of V that complete before s (there are $n-t$ of them) (see the striped area in Figure 6). We can thus bound the area above the work line (and the value of an optimal schedule of S_G) in terms of the size of a maximum edge biclique of G.

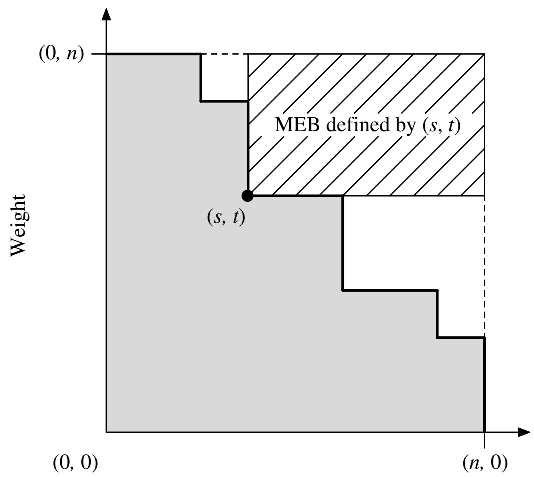

Figure 6. A two-dimensional Gantt chart representation of a schedule.

Formalizing the above-mentioned intuition, we obtain the following result.

Lemma 9.1. *Let* $\mathrm{val}(\sigma^*)$, *denote the value of an optimal schedule* σ^* *of* S_G. *If a maximum edge biclique of* G *has value* an^2 *for some* $a \in (0, 1]$, *then*

$$n^2 - an^2(\ln 1/a + 2) \le \mathrm{val}(\sigma^*) \le n^2 - an^2.$$

Proof. We start by showing that $\mathrm{val}(\sigma^*) \le n^2 - an^2$. Let $A \subseteq U, B \subseteq V$ be an edge biclique with value $|A| \cdot |B| = an^2$. Consider a feasible schedule σ that schedules the jobs in the order $U \backslash A \to B \to A \to V \backslash B$. The existence of such a schedule can be seen by observing that there is no precedence constraints from the jobs in A to the jobs in B. The bound now follows because $\mathrm{val}(\sigma^*) \le \mathrm{val}(\sigma)$ and

$$\mathrm{val}(\sigma) \le |U \backslash A| \cdot |B| + |U| \cdot |V \backslash B| = (n - |A|)|B| + n(n - |B|) = n^2 - |A \parallel B| = n^2 - an^2.$$

To prove the lower bound $n^2 - an^2(\ln 1/a + 2) \le \mathrm{val}(\sigma^*)$, we shall use $\sigma^*(i)$ to denote the total number of jobs of V scheduled before i jobs of U have been scheduled in σ^*. With this notation, the value of σ^* (where we let $\sigma^*(n+1) = n$) is

$$\sum_{i=1}^{n} (\sigma^*(i+1) - \sigma^*(i)) i = n^2 - \sum_{i=1}^{n} \sigma^*(i).$$

Note that, in any point of the schedule σ^*, the set of jobs of U that have not been scheduled (say, A) has no precedence constraints to the set of jobs of V that have been scheduled (say, B). It follows that A and B form an edge biclique of G with value $|A \parallel B|$. Because a maximum edge biclique of G has value $a \cdot n^2$, we have that $\sigma^*(i)(n - i + 1) \le an^2$ for $i = 1, \ldots, n$. Moreover, because $|V| \le n$, we have that $\sigma^*(i) \le n$ for $i = 1, \ldots, n$. Using these bounds on $\sigma^*(i)$, it follows that

$$\begin{aligned} n^2 - \sum_{i=1}^{n} \sigma^*(i) &= n^2 - \sum_{i=1}^{(1-a)n} \sigma^*(i) - \sum_{i=(1-a)n+1}^{n} \sigma^*(i) \\ &\ge n^2 - an^2 \sum_{i=1}^{(1-a)n} \frac{1}{n - i + 1} - \sum_{i=(1-a)n+1}^{n} n \\ &\ge n^2 - an^2(H_n - H_{\lfloor an \rfloor}) - an^2. \end{aligned}$$

The statement now follows by the bounds $\ln(n) \le H_n \le \ln(n) + 1$ on the harmonic series. □

We can now use hardness results for maximum edge biclique to obtain hardness results for $1 \mid \mathrm{prec} \mid \sum w_j C_j$. The best-known hardness result for MEB is from Ambühl et al. [4]. For our purposes, it will be convenient to state it as follows (the statement is obtained by using the standard method of graph products; see, e.g., Svensson [39, §4.5]).

THEOREM 9.1. *Let $\epsilon > 0$ be an arbitrarily small constant. There exist positive constants b and ϵ' (that depend on ϵ) so that for all constants $k > 0$, given an SAT instance ϕ of size N, we can probabilistically construct an n-by-n bipartite graph G in time $2^{O(N^\epsilon)}$ such that, with high probability,*

- (*completeness*) *if ϕ is satisfiable, then G has an edge biclique of value at least $(b+\epsilon')^k n^2$, and*
- (*soundness*) *if ϕ is not satisfiable, then all edge bicliques of G have value less than $b^k n^2$.*

By combining Theorem 9.1 with the bounds of Lemma 9.1, we have that, in the completeness case, S_G has a schedule of value at most

$$n^2(1-(b+\epsilon')^k),$$

whereas, in the soundness case, all schedules of S_G have value at least

$$n^2(1-b^k(\ln(1/b^k)+2)).$$

Clearly, there is a sufficiently large k (that depends on b and ϵ', which, in turn, depend on ϵ) such that

$$(b+\epsilon')^k > b^k(\ln(1/b^k)+2).$$

It follows that $1\,|\,\text{prec}\,|\sum w_j C_j$ has no PTAS unless SAT can be solved by a (probabilistic) algorithm that runs in time $2^{O(N^\epsilon)}$, where N is the instance size and $\epsilon > 0$ can be made arbitrarily close to 0.

THEOREM 9.2. *If there is a PTAS for $1\,|\,\text{prec}\,|\sum w_j C_j$, then SAT can be solved by a (probabilistic) algorithm that runs in time 2^{n^ε}, where $\varepsilon > 0$ can be chosen to be an arbitrarily small constant.*

Acknowledgments. The authors thank Andreas Schulz for many helpful discussions during his visit to IDSIA. The authors also thank two anonymous reviewers for providing extensive lists of comments that greatly improved this paper. Different parts of this work have appeared in preliminary form in APPROX'06, IPCO'07, and FOCS'07. This research is supported by Swiss National Science Foundation Projects 200021-104017/1 entitled *Power Aware Computing*, 200020-122110/1 entitled *Approximation Algorithms for Machine Scheduling Through Theory and Experiments III*. The first author is supported by Nuffield Foundation Grant NAL32608. Research by the third author was partly supported by the Swiss National Science Foundation Grant PBTIP1-135850 and research by the fourth author was partially supported by ERC Grant 228021-ECCSciEng.

Appendix. Realizer of convex bipartite posets. We create a realizer of size 3 for a given convex bipartite poset. In the sequel, we sometimes stress that a job j_i is a plus or minus job by writing j_i^+ and j_i^-, respectively. We also assume, without loss of generality, that the plus jobs are numbered such that $i < j$ if and only if $l(i) \le l(j)$ (breaking ties arbitrarily), where $j_i, j_j \in J^+$.

Given a convex bipartite poset $\mathbf{P} = (N, P)$, we partition its incomparable pairs into three sets E_1, E_2, and E_3 (also depicted in Figure A.1). A pair of incomparable jobs $(j_i, j_j) \in \text{inc}(\mathbf{P})$ is a member of

E_1 *if* $i > j$ and $j_i, j_j \in J^-$; *else, if* $i < j$ and $j_i, j_j \in J^+$; *else, if* $j_i \in J^-$ and $j_j \in J^+$.
E_2 *if* $i < j$ and $j_i, j_j \in J^-$; *else, if* $j_i \in J^+$, $j_j \in J^-$ and there exists a $k > i$ such that $(j_j, j_k) \in P$.
E_3 *if* $i > j$ and $j_i, j_j \in J^+$; *else, if* $j_i \in J^+$, $j_j \in J^-$ and $(j_j, j_k) \notin P$ for all $k > i$.

The following lemma is a direct consequence of the definition of E_1, E_2, and E_3.

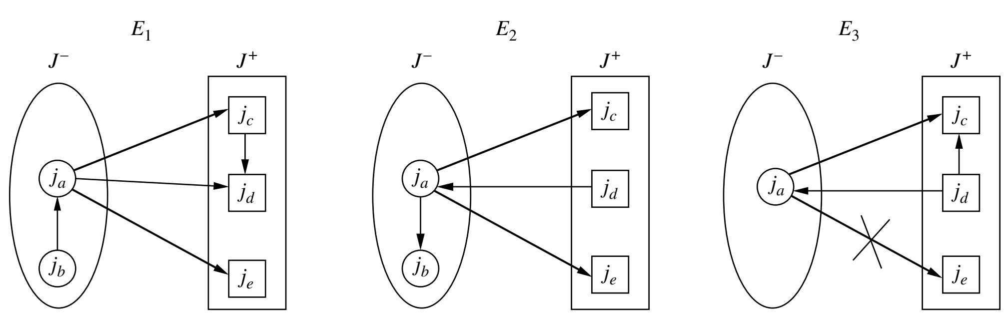

FIGURE A.1. The round and square nodes correspond to minus jobs and plus jobs, respectively.
Note. Bold edges correspond to precedence constraints, whereas the other edges are between incomparable jobs. In this example, we assume that $a < b$ and $c < d < e$.

Lemma A.1. *Let* **P** *be a convex bipartite order. Then,*
(i) *the sets* E_1, E_2, *and* E_3 *form a partition of* $inc(\mathbf{P})$, *and*
(ii) *for every* $(i, j) \in inc(\mathbf{P})$, *if* $(i, j) \in E_k$, *then* $(j, i) \notin E_k$, *where* $k \in \{1, 2, 3\}$.

Lemma A.2. *Let* $\bar{E}_1 = E_1 \cup P$, $\bar{E}_2 = E_2 \cup P$, *and* $\bar{E}_3 = E_3 \cup P$. *Then,* $\bar{E}_1, \bar{E}_2$, *and* $\bar{E}_3$ *are extensions of* P.

Proof. By the definition of $\bar{E}_i$, it follows that, if $(j_i, j_j) \in P$, then $(j_i, j_j) \in \bar{E}_i$, where $i = 1, 2, 3$. Moreover, it is easy to see (from Figure A.1) that the sets $\bar{E}_1$ and $\bar{E}_3$ do not contain cycles, i.e., are extensions of P.

Now, suppose $\bar{E}_2$ contains an alternating cycle C, i.e., it is a nonvalid extension. By the definition of E_2, we have $C \cap P \neq \varnothing$ and thus $C \cap (J^+ \times J^-) \neq \varnothing$. Let $j_i^- \in J^-$ be the minus job with largest index in the cycle, i.e., there does not exist a $k > i$ such that $j_k \in J^-$ is part of the cycle. Then, $(j_i^-, j_j^+) \in P \cap C$ and $(j_j^+, j_m^-) \in C$ for some jobs $j_j \in J^+$ and $j_m \in J^-$, where $m < i$. However, this implies that there exists an $n > j$ such that $(j_m^-, j_n^+) \in P$ (recall the definition of E_2). Together with convexity and the numbering of plus jobs, this implies $(j_m^-, j_j^+) \in P$, which contradicts the existence of $(j_j^+, j_m^-) \in C$. □

Let L_1, L_2, and L_3 be any linear extensions of $\bar{E}_1, \bar{E}_2$, and $\bar{E}_3$, respectively. That $\mathcal{R} = \{L_1, L_2, L_3\}$ is a realizer follows from the facts that all incomparable pairs are reversed (Lemma A.1) and that $\bar{E}_1, \bar{E}_2$, and $\bar{E}_3$ are valid extensions of **P** (Lemma A.2). Furthermore, all steps involved in creating $\mathcal{R}$ can clearly be accomplished in polynomial time.

We end by noting that the resulting upper bound of 3 on the dimension is indeed tight because a bipartite order **P** is 2-dimensional if and only if it is a strong bipartite order (Möhring [28]).

References

[1] Alimonti, P., V. Kann. 2000. Some APX-completeness results for cubic graphs. *Theoret. Comput. Sci.* **237**(1–2) 123–134.
[2] Ambühl, C., M. Mastrolilli. 2009. Single machine precedence constrained scheduling is a vertex cover problem. *Algorithmica* **53**(4) 488–503.
[3] Ambühl, C., M. Mastrolilli, O. Svensson. 2006. Approximating precedence-constrained single machine scheduling. J. Dray, K. Jansen, J. D. P. Rolim, U. Zwick, eds. *Proc. 9th Internat. Workshop Approximation Algorithms Combin. Optim. Problems (APPROX2006), Approximation, Randomization, and Combinatorial Optimization Algorithms and Techniques*, Lecture Notes in Computer Science, Vol. 4110. Springer, Berlin, 15–26.
[4] Ambühl, C., M. Mastrolilli, O. Svensson. 2007. Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constraint scheduling. *Proc. 48th Annual IEEE Sympos. Foundations Comput. Sci.*, IEEE, Washington, DC, 329–337.
[5] Ambühl, C., M. Mastrolilli, N. Mutsanas, O. Svensson. 2007. Scheduling with precedence constaints of low fractional dimension. M. Fischetti, D. P. Williamson, eds. *Proc. 12th Internat. IPCO Conf. (IPCO2007), Integer Programming and Combinatorial Optimization*, Lecture Notes in Computer Science, Vol. 4513. Springer, Berlin, 130–144.
[6] Bansal, N., S. Khot. 2009. Optimal long-code test with one free bit. *Proc.* 50*th Annual IEEE Sympos. Foundations Comput. Sci.* (*FOCS*), IEEE, Washington, DC, 453–462.
[7] Brightwell, G. R., E. R. Scheinerman. 1992. Fractional dimension of partial orders. *Order* **9**(2) 139–158.
[8] Chekuri, C., R. Motwani. 1999. Precedence constrained scheduling to minimize sum of weighted completion times on a single machine. *Discrete Appl. Math.* **98**(1–2) 29–38.
[9] 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**(5) 199–204.
[10] Correa, J. R., A. S. Schulz. 2005. Single machine scheduling with precedence constraints. *Math. Oper. Res.* **30**(4) 1005–1021.
[11] Eastman, W. L., S. Even, I. M. Isaacs. 1964. Bounds for the optimal scheduling of *n* jobs on *m* processors. *Management Sci.* **11**(2) 268–279.
[12] Felsner, S., W. T. Trotter. 1994. On the fractional dimension of partially ordered sets. *Discrete Math.* **136**(1-3) 101–117.
[13] Felsner, S., W. T. Trotter. 2000. Dimension, graph and hypergraph coloring. *Order* **17**(2) 167–177.
[14] Garey, M. R., D. S. Johnson, L. J. Stockmeyer. 1976. Some simplified NP-complete graph problems. *Theoret. Comput. Sci.* **1**(3) 237–267.
[15] Goemans, M. X., D. P. Williamson. 2000. Two-dimensional Gantt charts and a scheduling algorithm of Lawler. *SIAM J. Discrete Math.* **13**(3) 281–294.
[16] Graham, R., E. Lawler, J. K. Lenstra, A. H. G. Rinnooy Kan. 1979. Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, Vol. 5. North-Holland, Amsterdam, 287–326.
[17] Hall, L. A., D. B. Shmoys, J. Wein. 1996. Scheduling to minimize average completion time: Off-line and on-line algorithms. *Sympos. Discrete Algorithms* (*SODA*), Vol. 7. Society of Industrial and Applied Mathematics, Philadelphia, 142–151.
[18] 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**(3) 513–544.
[19] Hegde, R., K. Jain. 2007. The hardness of approximating poset dimension. *Electronic Notes Discrete Math.* **29** 435–443.
[20] Hochbaum, D. S. 1983. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Appl. Math.* **6**(3) 243–254.
[21] Khot, S. 2002. On the power of unique 2-prover 1-round games. *Proc.* 34*th Annual ACM Sympos. Theory Comput.* (*STOC*), ACM, New York, 767–775.
[22] Kolliopoulos, S. G., G. Steiner. 2002. Partially-ordered knapsack and applications to scheduling. *Proc.* 10*th Annual Eur. Sympos. Algorithms* (*ESA*), Springer, Berlin, 612–624.
[23] Lawler, E. L. 1978. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Ann. Discrete Math.* **2** 75–90.

[24] Lawler, E. L., J. K. Lenstra, A. H. G. Rinnooy Kan, D. B. Shmoys. 1993. Sequencing and scheduling: Algorithms and complexity. S. C. Graves, A. H. G. Rinnooy Kan, P. Zipkin, eds. *Handbooks in Operations Research and Management Science*, Vol. 4. North-Holland, Amsterdam, 445–552.

[25] Lenstra, J. K., A. H. G. Rinnooy Kan. 1978. The complexity of scheduling under precedence constraints. *Oper. Res.* **26**(1) 22–35.

[26] Ma, T.-H., J. P. Spinrad. 1994. On the 2-chain subgraph cover and related problems. *J. Algorithms* **17**(2) 251–268.

[27] Margot, F., M. Queyranne, Y. Wang. 2003. Decompositions, network flows, and a precedence-constrained single-machine scheduling problem. *Oper. Res.* **51**(6) 981–992.

[28] Möhring, R. H. 1989. Computationally tractable classes of ordered sets. L. Rival, ed. *Algorithms and Order*. Kluwer Academic, Dordrecht, The Netherlnads, 105–193.

[29] Nemhauser, G. L., L. E. Trotter. 1975. Vertex packings: Structural properties and algorithms. *Math. Programming* **8**(1) 232–248.

[30] Papadimitriou, C. H., M. Yannakakis. 1979. Scheduling interval-ordered tasks. *SIAM J. Comput.* **8**(3) 405–409.

[31] Pisaruk, N. N. 2003. A fully combinatorial 2-approximation algorithm for precedence-constrained scheduling a single machine to minimize average weighted completion time. *Discrete Appl. Math.* **131**(3) 655–663.

[32] Potts, C. N. 1980. An algorithm for the single machine sequencing problem with precedence constraints. *Math. Programming Stud.* **13** 78–87.

[33] Rabinovitch, I. 1978. The dimension of semiorders. *J. Combin. Theory Ser. A* **25** 50–61.

[34] Scheinerman, E. R., D. H. Ullman. 1997. *Fractional Graph Theory*. John Wiley and Sons, New York.

[35] Schulz, A. S. 1996. Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds. *Proc. 5th Conf. Integer Programming Combin. Optim.* (*IPCO*), Lecture Notes in Computer Science, Vol. 1084. Springer, Berlin, 301–315.

[36] Schuurman, P., G. J. Woeginger. 1999. Polynomial time approximation algorithms for machine scheduling: Ten open problems. *J. Scheduling* **2**(5) 203–213.

[37] Sidney, J. B. 1975. Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Oper. Res.* **23**(2) 283–298.

[38] Spencer, J. 1971. On minimum scrambling sets of simple orders. *Acta Mathematica* **22**(3–4) 349–353.

[39] Svensson, O. 2009. *Approximability of Some Classical Graph and Scheduling Problems*. Ph.D. thesis, Università della Svizzera Italiana, University of Lugano, Lugano, Switzerland.

[40] Svensson, O. 2010. Conditional hardness of precedence constrained scheduling on identical machines. *Proc. 42nd ACM Sympos. Theory Comput.* (*STOC*), ACM, New York, 745–754.

[41] Trotter, W. T. 1992. *Combinatorics and Partially Ordered Sets: Dimension Theory*. Johns Hopkins Series in the Mathematical Sciences, The Johns Hopkins University Press, Baltimore.

[42] Trotter, W. T. 1997. New perspectives on interval orders and interval graphs. R. A. Bailey, ed. *Surveys in Combinatorics*, Vol. 241. Mathematical Society Lecture Note Series, London, 237–286.

[43] Uhan, N. A. 2008. *Algorithmic and Game-Theoretic Perspectives on Scheduling*. Ph.D. thesis, Massachusetts Institute of Technology, Cambridge, MA.

[44] Woeginger, G. J. 2003. On the approximability of average completion time scheduling under precedence constraints. *Discrete Appl. Math.* **131**(1) 237–252.

[45] Yannakakis, M. 1982. On the complexity of partial order dimension problem. *SIAM J. Algebraic Discrete Methods* **22**(3) 351–358.