On the Approximability of Single Machine Scheduling with Precedence Constraints

Christoph Ambühl
Dep. of CS, The University of Liverpool, Ashton Building, Ashton Street, Liverpool L69 3BX, UK
email: christoph@csc.liv.ac.uk  http://www.csc.liv.ac.uk/~christoph/

Monaldo Mastrolilli
IDSIA, Galleria 2,6928 Manno-Lugano, Switzerland
email: monaldo@idsia.ch  http://www.idsia.ch/~monaldo/

Nikolaus Mutsanas
IDSIA, Galleria 2,6928 Manno-Lugano, Switzerland
email: nikos@idsia.ch  http://www.idsia.ch/~nikos/

Ola Svensson
CSC, KTH, SE-100 44 Stockholm, Sweden
email: osven@kth.se  http://www.nada.kth.se/~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 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 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

1. Introduction

The problem we consider in this paper is a classical problem in scheduling theory, known as \([prec]\sum w_jC_j\) in standard scheduling notation (see e.g. Graham et al. [14]). It is defined as the problem of scheduling a set \(N = \{1, \ldots, n\}\) of \(n\) jobs on a single machine, which 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) \(P = (N, P)\), 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-preemptive schedule which 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 shown to be strongly \(NP\)-hard already in 1978 by Lawler [21] and Lenstra & Rinnooy Kan [23]. For the general version of \([prec]\sum w_jC_j\), closing the approximability gap has been listed as one of ten outstanding open problems in scheduling theory (see e.g. [34]). On the positive side, several polynomial time 2-approximation algorithms are known. Pisarski [29] claims to have obtained the first such algorithm. Schulz [33] and Hall, Schulz, Shmoys & Wein [15] gave 2-approximation algorithms using linear programming relaxations. Chudak & Hochbaum [7] gave another algorithm based on a re-

1Different parts of this work have appeared in preliminary form in APPROX’06, IPCO’07 and FOCS’07.
2Note that [15] is a joint journal version of Schulz [33] and Hall, Shmoys & Wein [16] 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 class of algorithms. Indeed, in his Dissertation, Uhan [41] gave an instance for which any solution following the Sidney decomposition performs arbitrarily bad for the variable cost.

On the negative side, Woeginger [42] 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 [21] gave an exact algorithm for series-parallel orders already in 1978 (other classes of polynomially solvable instances are also known and the interested reader is referred to [22] for a survey). For interval orders and convex bipartite precedence constraints, Woeginger [42] gave approximation algorithms with approximation ratio arbitrarily close to the golden ratio \( \frac{\sqrt{5} - 1}{2} \approx 1.61803 \).

Recently, Ambühl & Mastrolilli [2] settled an open problem first raised by Chudak & Hochbaum [7] and whose answer was subsequently conjectured by Correa & Schulz [8]. In [8], the authors show that a positive answer of an open question raised in [7] would imply that 1|\( \text{prec} \]|\( \sum w_jC_j \) 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 [8] which in turn improved upon a previous approximation algorithm with a guarantee of \( (1 + \sqrt{5})/2 + \varepsilon \) [20]). More precisely, the results in [8, 2] prove that every instance \( S \) of 1|\( \text{prec} \]|\( \sum w_jC_j \) can be translated in polynomial time into a weighted graph \( G_S^S \), such that finding an optimum of \( S \) can be reduced to finding an optimum vertex cover in \( G_S^S \). Furthermore, this result even holds for approximate solutions: Finding an \( \alpha \)-approximate solution for \( S \) can be reduced to finding an \( \alpha \)-approximate vertex cover in \( G_S^S \). 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 provides an approximation guarantee of 2 already for the variable cost, in contrast to previously known algorithms, as mentioned above.

In this paper we continue to investigate the structure of 1|\( \text{prec} \]|\( \sum w_jC_j \). We point out an interesting relationship between the dimension theory of partial orders and 1|\( \text{prec} \]|\( \sum w_jC_j \). More specifically, in Section 3 we show that the vertex cover graph \( G_S^S \) associated with 1|\( \text{prec} \]|\( \sum w_jC_j \) is exactly the graph of incomparable pairs \( G_P^S \) defined in dimension theory [11]. This equivalence allows us to benefit from dimension theory. In particular, the chromatic number of \( G_P^S \) is at most \( d \), whenever the dimension of the poset at hand is (at most) \( d \). Hochbaum [17] showed that if a given graph for the vertex cover problem can be colored with \( d \) colors in polynomial time, then there exists a \( (2 - \frac{1}{d}) \)-approximation algorithm for the corresponding weighted vertex cover problem. It follows that there exists a \( (2 - \frac{1}{d}) \)-approximation algorithm for 1|\( \text{prec} \]|\( \sum w_jC_j \) 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 (Section 4.1) and semi-orders (Section 4.2), for which we exhibit 4/3-approximation algorithms that improve previous results by Woeginger [42].

Unfortunately, the framework described above fails in the case of interval orders (in this case the dimension can be of order \( \log \log n \) [39]). To overcome this difficulty, we further generalize this framework such that it can be applied to precedence constraints of low fractional dimension [5] (Section 5). The extended framework yields \((2 - \frac{1}{f})\)-approximation algorithms whenever precedence constraints have fractional dimension bounded by a constant \( f \) and satisfy an additional mild condition (Section 5). Since the fractional dimension of interval orders is bounded by 4 (Section 6.1), this gives a 3/2-approximation
algorithm and improves the previous result in [42]. The extended framework can also be applied to posets of interval dimension 2 (Section 6.2), bounded degree posets (Section 6.3), and posets obtained by lexicographic sums (Section 6.4).

In summary, the above results indicate a strong relationship between the approximability of \(1|\text{prec}| \sum w_j C_j\) and the fractional dimension \(\dim(P)\) of the precedence constraints. In particular, \(1|\text{prec}| \sum w_j C_j\) is polynomially solvable for \(f = 2\) [8, 2] but NP-hard already 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 [23], and the fractional dimension of these posets is bounded by 3 [10]. 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 Section 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. [25]). The reduction heavily relies on the connection between \(1|\text{prec}| \sum w_j C_j\) and weighted vertex cover described in [8, 2].

For the general problem \(1|\text{prec}| \sum w_j C_j\) with arbitrary partial orders we show in Section 8 that approximating the variable cost of \(1|\text{prec}| \sum w_j C_j\) is equivalent to approximating the vertex cover problem. This implies that a better than 2-approximation algorithm for \(1|\text{prec}| \sum w_j C_j\) 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 \(1|\text{prec}| \sum w_j C_j\), 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 Section 9 we establish a connection between \(1|\text{prec}| \sum w_j C_j\) 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 towards closing the approximability gap for this scheduling problem, a prominent problem in scheduling theory (see [34]). Subsequent to our work, Bansal & Khot [4] showed that the gap indeed closes assuming a variant of the unique games conjecture [19], by providing a 2-inapproximability result based on that assumption. Furthermore, this year the fourth author [37] established an interesting relationship between \(1|\text{prec}| \sum w_j C_j\) and the problem \(P|\text{prec}|C_{\text{max}}\), by showing that a \((2 - \epsilon)\)-hardness for \(1|\text{prec}| \sum w_j C_j\) would imply essentially the same hardness result for \(P|\text{prec}|C_{\text{max}}\), and would thus settle another prominent open question listed in [34]. This underscores the importance of understanding the approximability of \(1|\text{prec}| \sum w_j C_j\) even more.

2. Definitions and Preliminaries

2.1 Posets and Fractional Dimension Let \(P = (N, P)\) 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 inc(P) = \(\{ (x, y) \in N \times N : x\parallel y\ \text{in} P \}\) the set of incomparable pairs of \(P\) (note that, since incomparability is a symmetric relation, if \((x, y) \in \text{inc}(P)\) then \((y, x) \in \text{inc}(P)\) as well). A poset \(P\) 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., \(\text{inc}(P) = \emptyset\). A partial order \(P^0\) on \(N\) is an extension of a partial order \(P\) on the same set \(N\), if \(P \subseteq P^0\). An extension of \(P\) that is a linear order is called a linear extension of \(P\). Mirroring the definition of the fractional chromatic number of a graph, Brightwell & Scheinerman [5] introduce the notion of fractional dimension of a poset. Let \(F = \{L_1, L_2, \ldots, L_t\}\) be a nonempty multiset of linear extensions of \(P\). The authors in [3] call \(F\) a \(k\)-fold realizer of \(P\) if for each incomparable pair \((x, y)\), there are at least \(k\) linear extensions in \(F\) which 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 \((x, y) \in \text{inc}(P)\) a critical pair if for all \(z, w \in N \setminus \{x, y\}\) \(z < x\) 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 \(F\) which reverse the pair \((x, y)\) then \(F\) is a \(k\)-fold realizer of \(P\) and vice versa [5]. The fractional dimension of \(P\) is the smallest rational number \(\text{fdim}(P) \geq 1\) for which there exists a \(k\)-t-realizer of \(P\) so that \(t/k \leq \text{fdim}(P)\). Using this terminology, the dimension of \(P\), denoted by \(\dim(P)\), is the least \(t\) for which there exists a 1-fold realizer of \(P\). It is immediate that \(\text{fdim}(P) \leq \dim(P)\) for any poset \(P\). Furthermore, \(\text{fdim}(P) = 1\), or \(\text{fdim}(P) \geq 2\) [5].
2.2 The Scheduling Problem and Vertex Cover  In a series of recent papers [7, 8, 2] it was proved that 1|\text{prec}| \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 \([30, 7, 8]\) of 1|\text{prec}| \sum w_j C_j, using linear ordering variables \( \delta_{ij} \). The variable \( \delta_{ij} \) has value 1 if job \( i \) precedes job \( j \) in the corresponding schedule, and 0 otherwise. Correa & Schulz [8] proposed the following relaxation of 1|\text{prec}| \sum w_j C_j:

\[
[CS-IP] \quad \min \sum_{i \neq j} \delta_{ij} p_i w_j + \sum_{j \in S} p_j w_j + \sum_{(i,j) \in P} p_i w_j \\
\text{s.t.} \quad \delta_{ij} + \delta_{ji} \geq 1 \quad i \parallel j, \\
\delta_{ik} + \delta_{kj} \geq 1 \quad (i,j) \in P, i \parallel k \text{ and } k \parallel j, \\
\delta_{il} + \delta_{kj} \geq 1 \quad (i,j), (k,\ell) \in P, i \parallel \ell \text{ and } j \parallel k, \\
\delta_{ij} \in \{0,1\} \quad i \parallel j.
\]

Note that [CS-IP] can be interpreted as the problem of finding a minimum weighted vertex cover in an undirected graph [8]: given a scheduling instance \( S \) with precedence constraints \( P = (N, P) \), let \( G_p^S \) 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 \( (k,\ell) \) 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 & Schulz [8] 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 [8] was recently settled in the affirmative by Ambühl & Mastrolilli [2], even for approximate solutions. To summarize, the following theorem was proven.

**Theorem 2.1** ([2, 8]) Let \( S \) be an instance of 1|\text{prec}| \sum w_j C_j, and let \( G_p^S \) be the corresponding vertex cover graph. An \( \alpha \)-approximate solution to \( G_p^S \) can in polynomial time be turned into an \( \alpha \)-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 \( G_p^S \). We refer the interested reader to [8, 2] for a more comprehensive discussion.

We already mentioned that Hochbaum [17] 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 the above with the following observation.

**Observation 2.2** ([2, 8, 17]) If the graph \( G_p^S \) 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 soon give examples of precedence constraints for which \( G_p^S \) can be colored with few colors, after showing 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 \( P = (N, P) \) be any poset that is not a linear order. Felsner and Trotter [11] associate with \( P \) a hypergraph \( H_P \), called the hypergraph of incomparable pairs, defined as follows. The vertices of \( H_P \) are the incomparable pairs in \( P \). The edge set consists of those minimal (in terms of inclusion) sets \( U \) of incomparable pairs such that no linear extension of \( P \) reverses all incomparable pairs in \( U \). Let \( G_P \) denote the ordinary graph, called the graph of incomparable pairs, determined by all edges of size 2 in \( H_P \). We recall that the chromatic number of a hypergraph \( H = (V, \mathcal{E}) \), denoted \( \chi(H) \), is the least positive

---

\[ \text{Note that, while the poset } P \text{ is part of the definition of } S \text{ and could therefore be omitted, we use this notation to adhere to the notation of the graph of incomparable pairs in dimension theory (see Section 3). Thus, } P \text{ defines the structure of the graph } G_p^S, \text{ and the rest of } S \text{ the weights of its vertices.} \]
integer $t$ for which there is a function $f : V \to \{1, \ldots, t\}$ so that there is no $\alpha \in \{1, \ldots, t\}$ for which there is an edge $E \in E$ with $f(x) = \alpha$ for every $x \in E$. The following result due to Felsner & Trotter [11] associates a poset $P$ with $\mathcal{H}_P$ so that the dimension of $P$ is the chromatic number of $\mathcal{H}_P$.

**Proposition 3.1** ([11]) Let $P = (N, P)$ be a poset that is not a linear order. Then $\dim(P) = \chi(\mathcal{H}_P) \geq \chi(G_P)$.

Given a $k$-realizer $\mathcal{R} = \{L_1, L_2, \ldots, L_k\}$ of $P$, we can easily color $\mathcal{H}_P$ (and $G_P$) with $k$ colors: color vertex $(i, j)$ with some color $c$ whenever $(j, i) \in L_c$ (in case $(j, i)$ 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 $\mathcal{H}_P$. In [5], it was noted that the same relationship holds for the fractional versions, i.e., $\deltaim(P) = \chi_f(\mathcal{H}_P) \geq \chi_f(G_P)$, where $\chi_f(A)$ denotes the fractional chromatic number of $A$. We refer the reader to [32] 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 $G^S_P$ associated with $1|\text{prec}| \sum w_j C_j$ and the graph of incomparable pairs $G_P$ 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** ([11] [39]) Let $P = (N, P)$ be a poset that is not a linear order. Then the graph $G_P$ is bipartite if and only if $\dim(P) = 2$.

Using a different approach, Correa & Schulz [8] rediscovered the above theorem for the vertex cover graph $G_P^S$, independent of the connection pointed out by Proposition 3.2. Moreover, the following theorem follows easily from Observation 2.2 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 $(2 - \frac{2}{k})$-approximation algorithm.

A natural question is for which posets one can construct a $k$-realizer in polynomial time. Yannakakis [43] proved that determining whether the dimension of a poset is at most $k$ is NP-complete for every $k \geq 3$. Moreover, Jain & Hegde [18] 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 (Section 4.1) and semi-orders (Section 4.2), a minimal realizer can be computed in polynomial time.

Finally, by Proposition 3.1 we remark that $\dim(P)$ and $\chi(G_P)$ are, in general, not the same (see [11] for an example where $\dim(P)$ is exponentially larger than $\chi(G_P)$). However, it is an immediate consequence of Theorem 3.1 that $\dim(P) = \chi(G_P)$ when $\dim(P) = 3$. Therefore, a 3-realizer for a 3-dimensional partial order $P$ (as in Sections 4.1 and 4.2) immediately gives an optimal coloring for $G_P$.

### 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 semi-orders. We note that these results can be generalized to a richer class of posets obtained by the lexicographic sum of posets of the above types (as proved for the fractional dimension in Section 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* $P = (N = J^- \cup J^+, P)$ is defined as follows.
(i) The set of jobs are divided into two disjoint sets $J^- = \{j_1, \ldots, j_a\}$ and $J^+ = \{j_{a+1}, \ldots, j_n\}$, 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 $1 \leq l(k) \leq r(k) \leq n$ such that $(j_{l(k)}, j_{r(k)}) \in P$ 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. Lemma 4.1 states that the class of convex bipartite orders has dimension $\leq 3$ (this result was previously unknown, to the best of our knowledge).

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

The proof of this Lemma can be found in Appendix A. 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 which 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 Semi-Orders
A poset $P = (N, P)$ is a semi-order, also called unit interval order, if there is a function $F$, which 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 semi-orders, which allow arbitrary interval lengths in the above definition.

Semi-orders can be recognized in $O(n^2)$ time. Moreover, Rabinovitch proved, by constructing a realizer, that the dimension of semi-orders is at most three. 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 semi-order has a polynomial time $4/3$-approximation algorithm.

Finally, we show that the above approach cannot be applied to interval orders. The dimension of interval orders can be of order $\log \log n$. 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 $I_n$ denote the interval order determined by the set of all closed intervals with distinct integer endpoints. We fix it convenient to view the elements of $I_n$ as 2-element subsets of $[n]$ with $(\{i_1, i_2\}, \{i_3, i_4\})$ in $I_n$ 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 $\{I_n : n \geq 2\}$ are called canonical (we point out that canonical interval orders are also used to prove that interval orders have unbounded dimension, see e.g. 20).

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

**Proof.** The chromatic number $\chi(G_{I_n})$ is clearly a non-decreasing function of $n$. We assume that $\chi(G_{I_n}) \leq k$ for all $n \geq 2$ and obtain a contradiction when $n$ is sufficiently large.

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

Let $n_0$ equal the Ramsey number $R(3 : h_1, h_2, h_3, \ldots, h_k)$, where $h_1 = h_2 = \cdots = h_k = 4$. Now pick $n$ to be greater or equal to $n_0$ and hence $\lceil |n| \rceil \geq n_0$. Consider any coloring of $G_{I_n}$ and the corresponding

\footnote{Note that we can assume without loss of generality that $i < j < l$.}
map $\varphi$. By Ramsey’s Theorem \[39\], 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 $G_L$, which are colored by the same color. This contradicts the existence of a valid $k$-coloring for $G_L$ when $n \geq n_0$. \qed

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 Section 6.

We say that a poset $P$ admits an efficiently samplable $k$-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 $F = \{L_1, L_2, \ldots, L_t\}$ with probability $1/t$.

The following theorem generalizes Theorem 3.2. Its proof uses a similar argument as Hochbaum’s original proof of Theorem \[32\] 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 $k$-t-realizer, has a randomized $(2 - \frac{1}{k})$-approximation algorithm.

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

Let $G_P^S = (V_P, E_P)$ be the weighted vertex cover instance associated with $S$ where each vertex (incomparable pair) $(i, j) \in V_P$ has weight $w_{(i,j)} = p_i \cdot w_j$, as specified in Section 2.2. Using a result by Nemhauser & Trotter \[27\], the [CS-LP] formulation of $G_P^S$ can be solved combinatorially and in polynomial time, and has half-integral extreme point solutions. Furthermore, let $V_t$ be the set of vertices with value $i$ ($i = 0, \frac{1}{2}, 1$) in the optimum solution. Denote by $G_P^S[V_{1/2}]$ the subgraph of $G_P^S$ induced by the vertex set $V_{1/2}$. We consider the linear extensions of $F$ as outcomes in a uniform sample space. For an incomparable pair $(x, y)$, the probability that $y$ is over $x$ in $F$ is given by

$$\text{Prob}_F[y > x] = \frac{1}{t}|\{i = 1, \ldots, t : y > x \in L_i\}| \geq \frac{k}{t}$$

(4)

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

Let us pick one linear extension $L$ uniformly at random from $F = \{L_1, \ldots, L_t\}$. 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 $^5$

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

(5)

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

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

As observed in \[17\], $V_1 \cup C$ gives a valid vertex cover for the graph $G_P^S$. Moreover, the expected value of

\[5\]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 $(a, b) \in L$, and the incomparable pairs that are reversed in $L$, namely $(b, a) \notin L$, form the complement of this set.
the cover is bounded as follows

\[ E[w(V_1 \cup C)] \leq w(V_1) + \left(1 - \frac{k}{t}\right) w(V_{1/2}) \]

\[ \leq 2 \left(1 - \frac{k}{t}\right) \left(w(V_1) + \frac{1}{2}w(V_{1/2})\right) \]

\[ \leq 2 \left(1 - \frac{2}{k\sqrt{t}}\right) OPT \]

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

6. Precidence Constraints with Low Fractional Dimension

6.1 Interval orders A poset \( P = (N, P) \) is an interval order if there is a function \( F \), which 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 [28]. The dimension of interval orders can be of order \( \log \log n \) [39], whereas the fractional dimension is known to be less than 4 [7], and this bound is asymptotically tight [10]. 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 \( P = (N, P) \), 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.

**Theorem 6.1 (Rabinovitch [31])** A poset \( P = (N, P) \) 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 \( \mathcal{F} = \{L_1, \ldots, L_t\} \) 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 \( B = N \setminus A \) is over \( A \). Now let \( \mathcal{F} \) be the multiset of all the \( L_A \)'s. 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 \( 1|\text{Prec}||\sum w_j C_j \) when the precedence constraints form an interval order. The described algorithm can easily be derandomized by using the standard method of conditional probabilities.

**Theorem 6.2** Problem \( 1|\text{Prec}||\sum w_j C_j \) with precedence constraints forming an interval order has a deterministic polynomial time \( 3/2 \)-approximation algorithm.

6.2 Interval dimension two The interval dimension of a poset \( P = (N, P) \), denoted by \( \text{dim}_I(P) \), is defined [39] as the least \( t \) for which there exist \( t \) extensions \( Q_1, Q_2, \ldots, Q_t \), so that:

- \( P = Q_1 \cap Q_2 \cap \cdots \cap Q_t \)
- \((N, Q_i)\) is an interval order for \( i = 1, 2, \ldots, t \).

Obviously, if \( P \) is an interval order, then \( \text{dim}_I(P) = 1 \) and, in general, since a linear order is an interval order, \( \text{dim}_I(P) \leq \text{dim}(P) \).

The class of posets of interval dimension 2 forms a proper superclass of the class of interval orders. Posets of interval dimension two can be recognized in \( O(n^2) \)-time [24]. Given a poset \( P \) with \( \text{dim}_I(P) = 2 \), their algorithm also yields an interval realizer \( \{Q_1, Q_2\} \). As described in Section 6.1 we obtain \( k \)-fold realizers \( \mathcal{F}_1 = \{L_1, L_2, \ldots, L_t\} \) and \( \mathcal{F}_2 = \{L'_1, L'_2, \ldots, L'_t\} \) of \( Q_1 \) and \( Q_2 \), respectively, with \( k = 2^{n-2} \) and
Let $P = (N, P)$ be a poset. For any job $j \in N$, define the degree of $j$, denoted $\text{deg}(j)$, as the number of jobs comparable (but not equal) to $j$ in $P$. Given a job $j$, let $D(j)$ denote the set of all jobs which are less than $j$, and $U(j)$ those which are greater than $j$ in $P$. Define $\text{deg}_D(j) = |D(j)|$ and $\Delta_D(P) = \max\{\text{deg}_D(j) : j \in N\}$. The quantities $\text{deg}_U(j)$ and $\Delta_U(P)$ are defined analogously.

The NP-completeness proof for $1|\text{prec}| \sum w_i C_j$ given by Lenstra & Rinnoy Kan [23] was actually provided for posets $P$ with $\Delta_D(P) = 2$. 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 $1|\text{prec}| \sum w_i C_j$ has a polynomial time $(2 - 2/f)$-approximation algorithm, where $f = 1 + \max\{\min\{\Delta_D, \Delta_U\}, 1\}$.

**Proof.** Let $P = (N, P)$ be the poset representing the precedence constraints with bounded $\min\{\Delta_D, \Delta_U\}$. Assume, without loss of generality, that $P$ is not decomposable with respect to lexicographic sums (see Section 6.4). Otherwise, a decomposition with respect to lexicographic sums can be done in $O(n^2)$ time (see e.g. [23]), 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:

1. $x > (D(y) \cup \{y\})$ in $M$ if $|D(y)| < \Delta_D$
2. $x > D(y)$ in $M$ if $|D(y)| = \Delta_D$

In [10], Felsner & Trotter 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 $M = \{M_1, M_2, \ldots, M_t\}$ 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 $M_i \in M$, 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 [10] we obtain a $k$-fold realizer $F = \{L_1, \ldots, L_t\}$ of $P$ with $t = n!$ and $k = n!/(\Delta_D + 1)$. Moreover, we can efficiently pick uniformly at random one linear extension from $F$; 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 [10]. 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 $\Delta_U$ instead of $\Delta_D$. \hfill $\Box$

We remark that it is necessary to use the *fractional* dimension for obtaining the above result. To see this, consider the incidence poset $P(G) = (N, P)$ 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 $v \in \{v_1, v_2\}$. Since every edge is adjacent to only two vertices, $\Delta_D$ is bounded by 2. For the complete graph on $n$ nodes, $K_n$, Spencer [39] showed that $\text{dim}(P(K_n)) = \Theta(\log \log n)$ whereas from the above discussion we have $\text{fdim}(P(K_n)) \leq 1 + \min\{\Delta_U, \Delta_D\} \leq 3$.

### 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 here, lexicographic sums, comes from a very simple pictorial idea (see [39] for a more comprehensive discussion). Take a poset $P = (N, P)$ and replace

---

6 This is needed in order to be able to apply the algorithm of Felsner & Trotter [10] below.
each of its points \( x \in N \) with a partially ordered set \( Q_x \), 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 \( P = (N, P) \) and a family of posets \( S = \{ (Y_x, Q_x) \mid x \in N \} \) indexed by the elements in \( N \), the lexicographic sum of \( S \) 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 \( P \).

(ii) \( x_1 = x_2 \) and \( y_1 \leq y_2 \) in \( Q_{x_1} \).

We call \( P = P \cup S \) 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 non-trivial lexicographic sum.

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

**Theorem 6.5** Problem 1|\( prec | \sum w_j C_j \), whenever precedence constraints form a lexicographic sum whose components \( i \in P \) admit efficiently sampleable realizers, has a polynomial time randomized \((2 - \frac{2}{k})\)-approximation algorithm, where \( t/k = \max_{i \in P} (t_i/k_i) \).

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|\( prec| \sum w_j C_j \) 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|\( prec | \sum w_j C_j \): Finding an optimum solution to a scheduling instance \( S \), where precedence constraints are given by an interval order \( I \), is equivalent to solving the minimum weighted vertex cover problem in the graph \( G^S_I \) (see Section 2.3).

**Theorem 7.1** Problem 1|\( prec | \sum w_j C_j \) 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 [12]. We provide a reduction from the minimum vertex cover problem on graphs with bounded degree 3 to 1|\( prec | \sum w_j C_j \) with interval order precedence constraints.

Given a connected graph \( G = (V, E) \) with bounded degree 3, we construct an instance \( S \) of 1|\( prec | \sum w_j C_j \) with interval order precedence constraints so that the graph \( G^S_I \) has a weighted vertex cover with value less than \( m + c + 1 \) 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 \( T \) top-down and left-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 \( u_{i2}, u_{i1} \) and edges \( \{u_{i2}, u_{i1}\}, \{u_{i1}, v_i\} \). Furthermore, for each edge \( \{v_i, v_j\} \in E \setminus E_T \) with \( i < j \) we add vertices \( e_{i1}, e_{i2} \) and edges \( \{v_i, e_{i1}\}, \{e_{i1}, e_{i2}\}, \{e_{i2}, u_{i2}\} \).

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

**Claim 1.** Let \( C_s \subseteq V \) and \( C'_s \subseteq V' \) be optimum vertex cover solutions to \( G \) and \( G' \), respectively, then \( |C_s| = |C'_s| - |V| - |E \setminus E_T| \).

**Proof of Claim.** 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 \( |C| + |V| + |E \setminus E_T| \). In \( C' \) we include \( u_{i1} \) for all \( i \) with \( v_i \in V \setminus C \); \( u_{i2} \)
for all $i$ with $v_i \in C$; $e_{ij}^1$ for each $(v_i, v_j) \in E \setminus E_T$ with $v_i \in V \setminus C$; $e_{ij}^2$ for each $(v_i, v_j) \in E \setminus E_T$ 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 $v_i \notin C', v_j \notin C'$. Since $C'$ is a feasible vertex cover of $G'$ we have that $(v_i, v_j) \in E \setminus E_T$ and either $\{e_{ij}^1, e_{ij}^2, u_i^1\} \subseteq C'$ or $\{e_{ij}^1, u_i^1, u_j^1\} \subseteq C'$. Thus we can obtain a vertex cover $C'' \subseteq V'$ of $G'$ with $|C''| \leq |C'|$ by letting $C'' = (C' \setminus \{u_i^1, e_{ij}^1\}) \cup \{v_j, j\}$. Repeating this procedure will result in a vertex cover $C''' \subseteq V'$ of $G'$ with $|C'''| \leq |C'|$ such that $C = C''' \cap V$ is a feasible vertex cover of $G$. Furthermore it is easy to see that $|C| \leq |C'''| - |V| - |E \setminus E_T|$.

\[\square\]

**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 $I$ as defined below (see Figure 2 for an example), where $k$ is a large value to be determined later.

<table>
<thead>
<tr>
<th>Job</th>
<th>Interval Repr.</th>
<th>Proc. Time</th>
<th>Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>$s_0$</td>
<td>$[-1,0]$</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>$s_1$</td>
<td>$[0,1]$</td>
<td>$1/k^j$</td>
<td>$k^j$</td>
</tr>
<tr>
<td>$s_j, j = 2, \ldots,</td>
<td>V</td>
<td>$</td>
<td>$[i,j]$, where ${v_i, v_j} \in E_T, i &lt; j$</td>
</tr>
<tr>
<td>$m_i, i = 1, \ldots,</td>
<td>V</td>
<td>$</td>
<td>$[i - \frac{1}{2},</td>
</tr>
<tr>
<td>$e_i, i = 1, \ldots,</td>
<td>V</td>
<td>$</td>
<td>$[</td>
</tr>
<tr>
<td>$b_{ij},$ where ${v_i, v_j} \in E \setminus E_T, i &lt; j$</td>
<td>$[i, j - \frac{1}{2}]$</td>
<td>$1/k^j$</td>
<td>$k^j$</td>
</tr>
</tbody>
</table>

Remark 7.1 Let $i$ and $j$ be two jobs in $S$ with interval representations $[a,b]$ and $[c,d]$ respectively, where $a \leq d$. By the construction of the scheduling instance $S$ we have $p_i \leq 1/k^{|V|}$ and $w_j \leq k^{|V|}$. It follows that $p_i \cdot w_j = 1$ or $p_i \cdot w_j \leq 1/k$ if $i$ and $j$ are incomparable, since $p_i \cdot w_j \geq k$ 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 $|b| = |c|$.

Let $D = \{ (s_0, s_1) \}$

\[\cup \{(s_i, s_j) : v_i$ is the parent of $v_j$ in $T \}$

\[\cup \{(s_i, m_i), (m_i, e_i) : i = 1, 2, \ldots, |V| \}$

\[\cup \{(s_i, b_{ij}), (b_{ij}, m_j) : (v_i, v_j) \in E \setminus E_T, i < j \}$

By the interval representation of the jobs and the remark above, 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 $p_i \cdot w_j \leq 1/k$. 

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

![Graph Image](image-url)
Claim 3. Let $G'_I = (D, E_I)$ be the subgraph of $G'^S$ induced by the vertex subset $D$. Then $G'_I$ and $G'$ are isomorphic.

Proof of Claim. We relate the two graphs $G'_I$ and $G'$ by the bijection $f : D \rightarrow V'$, defined as follows.

\[
 f((a, b)) = \begin{cases} 
 v_j, & \text{if } (a, b) = (s_i, s_j), \\
 u_1^*, & \text{if } (a, b) = (s_i, m_i), \\
 u_2^*, & \text{if } (a, b) = (m_i, e_i), \\
 e_1^j, & \text{if } (a, b) = (s_i, b_{ij}), \\
 e_2^j, & \text{if } (a, b) = (b_{ij}, m_j).
\end{cases}
\]

Suppose $\{(a, b), (c, d)\} \in E_I$. By the fact that $I$ is an interval order (i.e., does not contain any $2 + 2$ structures, see Figure 3 as induced posets \cite{29}) together with the definition of $D$, we can assume that $b = c$ and $a \neq d$.

\[\begin{array}{c}
\circ \\
\circ
\end{array}\]

Figure 3: A $2 + 2$ poset.

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 $f((s_j, b_{jk})) = e_{k}^{ij}$ and by definition of $G'$ we have $\{v_j, e_{k}^{ij}\} \in E'$.

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 the two above and it is straightforward to check the implication $\{(a, b), (b, d)\} \in E_I \Rightarrow \{f((a, b)), f((b, c))\} \in E'$.

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 \( k < i < j \). Since \( s_k \)'s interval representation is completely to the left of \( s_j \)'s interval representation in I the incomparable pairs \( (s_k, s_i) \) and \( (s_i, s_j) \) cannot be reversed in the same linear extension, i.e., \( \{(s_k, s_i), (s_i, s_j)\} \in E_I \).

(a = e_{ij}^1, b = e_{ij}^2, i < j) Then \( f^{-1}(v_i) = (s_k, s_i) \) and \( f^{-1}(v_j) = (s_i, s_j) \) for some \( k < i < j \). Since \( s_k \)'s interval representation is completely to the left of \( b_{ij} \)'s interval representation in I the incomparable pairs \( (s_k, s_i) \) and \( (s_i, b_{ij}) \) cannot be reversed in the same linear extension, i.e., \( \{(s_k, s_i), (s_i, b_{ij})\} \in E_I \).

The remaining cases (a = e_{ij}^1, b = e_{ij}^2, i < j), (a = e_{ij}^1, 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 the two above and omitted.

We have thus proved that \( \{(a, b), (b, d)\} \in E_I \iff \{(f((a, b)), f((b, c)))\} \in E' \), i.e., the function \( f \) defines an isomorphism between \( G'_I \) 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 \( C_I \) and \( C'_I \) be minimum vertex covers of \( G^S_I \) and \( G'_I \) and denote their respective values by \( w(C_I) \) and \( w(C'_I) \). Since \( G'_I \) is unweighted we have \( w(C'_I) = |C'_I| \). By the selection of \( k \) and Claim 2, we have

\[
\sum_{(i,j)\in \text{inc}(I)} p_j w_j < 1, \quad \text{and thus } w(C'_I) = |w(C_I)| .
\]

Since \( G'_I \) and \( G' \) are isomorphic (Claim 3) we have by Claim 4 that graph \( G \) has an optimal vertex cover less at most \( m \) if and only if \( |w(C_I)| \leq m + c \), where

\[
e = |V| + |E \setminus E_T|. \tag{9}
\]

We remark that the above 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 \( G^S_I \) obtained from scheduling instances with interval order precedence constraints.

8. Hardness of Variable Cost

We show that approximating the variable cost of scheduling problem \( 1|\text{prec}| \sum w_j C_j \) is equivalent to approximating the vertex cover problem. Theorem 2.1 implies that minimizing the variable cost of \( 1|\text{prec}| \sum w_j C_j \) 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 \( 1|\text{prec}| \sum w_j C_j \) 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 \( 1/k^2 \) and 0, respectively. Conversely, the processing time and weight for a job \( v_i'' \) are 0 and \( k^2 \), respectively.

\( S \) has the following precedence constraints: for each edge \( \{v_i, v_j\} \in E \), the precedence constraints \( v_i' < v_j'' \) and \( v_i'' < v_j' \). 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 \( G^S_P \). It has at most \( n^2 \) vertices. The \( n \) vertices corresponding to the incomparable pairs \( (v_i', v_j'') \) have weight 1. All other vertices have weight at most \( 1/k^2 \), 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 \( (v_i', v_j'') \) and \( (v_j', v_j'') \) in \( G^S_P \) if and only if both precedence constraints \( v_i' \rightarrow v_j'' \) and \( v_j' \rightarrow v_j'' \) are present in \( S \). This in turn is the case if and only if \( \{v_i, v_j\} \in E \).
Figure 4: The transformation of a graph $G$.

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

We point out that the above 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 following 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 below. This relationship together with an inapproximability result for maximum edge biclique (MEB) yields Theorem 9.2 i.e., that the scheduling problem has no PTAS unless there is a probabilistic algorithm with running time $2^{N^{\epsilon}}$ 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 $P = U \times V \setminus E$. 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.

Figure 5: An example of a graph $G$ with its associated scheduling instance $S_G$.

The intuition behind the relationship between $1|\text{prec}|\sum w_j C_j$ and MEB is best seen by considering 2D Gantt charts, first introduced by Eastman et al. and later revived by Goemans and Williamson to give elegant proofs for various results related to $1|\text{prec}|\sum w_j C_j$. In a 2D Gantt chart, we have a horizontal axis of processing time and a vertical axis of weight. For a scheduling instance of the above 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 2D Gantt chart by placing the corresponding rectangles of the jobs in the order of the schedule such that the startpoint of a job is the endpoint 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 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: 2D Gantt chart representation of a schedule.](image)

Formalizing the above intuition we obtain the following result

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

\[
\begin{align*}
n^2 - an^2 \left(\ln \frac{1}{a} + 2\right) &\leq \text{val}(\sigma^\star) \leq n^2 - an^2.
\end{align*}
\]

**Proof.** We start by showing that \(\text{val}(\sigma^\star) \leq n^2 - an^2\). Let \(A \subseteq U, B \subseteq V\) be an edge biclique with value \(|A| \cdot |B| = an^2\). Consider a feasible schedule \(\sigma\) that schedules the jobs in the order \(U \setminus A \rightarrow B \rightarrow A \rightarrow V \setminus B\). 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 since \(\text{val}(\sigma^\star) \leq \text{val}(\sigma)\) and

\[
\begin{align*}
\text{val}(\sigma) &\leq |U \setminus A| \cdot |B| + |U| \cdot |V \setminus B| = (n - |A|)|B| + n(n - |B|) = n^2 - |A||B| = n^2 - an^2.
\end{align*}
\]

To prove the lower bound \(n^2 - an^2 \left(\ln \frac{1}{a} + 2\right) \leq \text{val}(\sigma^\star)\) we shall use \(\sigma^\star(i)\) to denote the total number of jobs of \(V\) scheduled before \(i\) jobs of \(U\) have been scheduled in \(\sigma^\star\). With this notation the value of \(\sigma^\star\) (where we let \(\sigma^\star(n+1) = n\)) is

\[
\begin{align*}
\sum_{i=1}^{n} (\sigma^\star(i+1) - \sigma^\star(i)) i &= n^2 - \sum_{i=1}^{n} \sigma^\star(i).
\end{align*}
\]

Note that in any point of the schedule \(\sigma^\star\), 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 $|A||B|$. As a maximum edge biclique of $G$ has value $a \cdot n^2$, we have that $\sigma^*(i)(n-i+1) \leq an^2$ for $i = 1, \ldots, n$. Moreover, since $|V| \leq n$ we have that $\sigma^*(i) \leq n$ for $i = 1, \ldots, n$. Using these bounds on $\sigma^*(i)$, it follows that

$$n^2 - \sum_{i=1}^{n} \sigma^*(i) = n^2 - \sum_{i=1}^{n} (1-a)n^2 - \sum_{i=(1-a)n+1}^{n} \sigma^*(i) \geq n^2 - an^2 \sum_{i=1}^{n} \frac{1}{n-i+1} - \sum_{i=(1-a)n+1}^{n} n \geq n^2 - an^2(H_n - H_{(an)}) - an^2.$$

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

We can now use hardness results for maximum edge biclique to obtain hardness results for $1|\text{prec}|\sum w_j C_j$. The best known hardness result for MEB are due to Ambühl et al. [3]. 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. Section 4.5 in [35]).

**Theorem 9.1** Let $\epsilon > 0$ be an arbitrarily small constant. There exist positive constants $b$ and $\epsilon'$ (that depend on $\epsilon$) so that for all constants $k > 0$, given a SAT instance $\phi$ 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 $\phi$ is satisfiable then $G$ has an edge biclique of value at least $(b + \epsilon')^k n^2$;
- (Soundness) if $\phi$ is not satisfiable then all edge bicliques of $G$ have value less than $b^k n^2$.

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

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

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

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

Clearly, there is a sufficiently large $k$ (that depends on $b$ and $\epsilon'$, which in turn depend on $\epsilon$) 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 $2^{n^\epsilon}$, where $\epsilon > 0$ can be chosen to be an arbitrarily small constant.

**Acknowledgments.** We are grateful to Andreas Schulz for many helpful discussions we had with him during his visit to IDSIA. We would like to thank two anonymous reviewers for providing extensive lists of comments that greatly improved this paper. This research is supported by Swiss National Science Foundation project 200021-104017/1, “Power Aware Computing”, and by the Swiss National Science Foundation project 200020-122110/1 “Approximation Algorithms for Machine Scheduling Through Theory and Experiments III”. The first author is supported by Nuffield Foundation Grant NAL32608. The third author is also supported by the ERC Advanced investigator grant 226203.

**References**


Appendix A. 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 \( l(i) \leq l(j) \) (breaking ties arbitrarily), where \( j_i, j_j \in J^+ \).

Given a convex bipartite poset \( P = (N, P) \), we partition its incomparable pairs into three sets \( E_1, E_2 \), and \( E_3 \) (also depicted in Fig. 7). A pair of incomparable jobs \( (j_i, j_j) \in \text{inc}(P) \) is a member of

<table>
<thead>
<tr>
<th>Condition</th>
<th>( E_1 )</th>
<th>( E_2 )</th>
<th>( E_3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( i &gt; j ) and ( j_i, j_j \in J^- )</td>
<td>\text{else if} ( i &lt; j ) and ( j_i, j_j \in J^+ ) \text{else if} ( j_i \in J^- ) and ( j_j \in J^+ ).</td>
<td>\text{if} ( i &lt; j ) and ( j_i, j_j \in J^- ) \text{else if} ( j_i \in J^+ ), ( j_j \in J^- ) and there exists a ( k &gt; i ) such that ( (j_j, j_k) \in P ).</td>
<td>\text{if} ( i &gt; j ) and ( j_i, j_j \in J^+ ) \text{else if} ( j_i \in J^+ ), ( j_j \in J^- ) and ( (j_j, j_k) \notin P ) for all ( k &gt; i ).</td>
</tr>
</tbody>
</table>

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

Lemma A.1  Let \( P \) be a convex bipartite order then
Figure 7: The round and square nodes correspond to minus-jobs and plus-jobs, respectively. Bold edges correspond to precedence constrains, whereas the other edges are between incomparable jobs. In this example we assume that $a < b$ and $c < d < e$.

(i) The sets $E_1, E_2,$ and $E_3$ form a partition of $\text{inc}(P)$;

(ii) For every $(i, j) \in \text{inc}(P)$, if $(i, j) \in E_k$ then $(j, i) \notin E_k$, where $k \in \{1, 2, 3\}$.

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

**Proof.** By the definition of $\tilde{E}_i$, it follows that if $(j_i, j_j) \in P$ then $(j_i, j_j) \in \tilde{E}_i$, where $i = 1, 2, 3$. Moreover, it is easy to see (Fig. 7) that the sets $\tilde{E}_1$ and $\tilde{E}_3$ do not contain cycles, i.e., are extensions of $P$.

Now suppose $\tilde{E}_2$ contains an alternating cycle $C$, i.e., it is a non valid extension. By the definition of $E_2$ we have $C \cap P \neq \emptyset$ and thus $C \cap (J^+ \times J^-) \neq \emptyset$. 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 $(j^-_n, j^+_m) \in P$ (recall the definition of $E_2$). Together with convexity and the numbering of plus-jobs this implies $(j^-_n, j^+_j) \in P$, which contradicts the existence of $(j^+_j, j^-_m) \in C$. □

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

We end by noting that the resulting upper bound of three on the dimension is indeed tight, since a bipartite order $P$ is 2-dimensional if and only if it is a strong bipartite order [26].