Precedence Constraint Scheduling and Connections to Dimension Theory of Partial Orders

Christoph Ambühl†, Monaldo Mastrolilli‡, Nikolaus Mutsanas‡, and Ola Svensson‡

†University of Liverpool, Great Britain, christoph@csc.liv.ac.uk

‡IDSIA, Switzerland, {monaldo,nikos,ola}@idsia.ch

Abstract

In this paper we survey recent results on the precedence constraint single machine scheduling problem to minimize the weighted sum of completion times. We show how we can benefit from known results in dimension theory of partial orders to obtain improved approximate solutions when precedence constraints have low fractional dimension. We believe that dimension theory of partial orders could be of interest for other linear ordering problems as well.

1 Introduction

The problem we consider in this paper is a classical problem in scheduling theory, known as $1|prec|\sum w_jC_j$ in standard scheduling notation (see e.g. Graham et al. [15]). A set $N = \{1, \dots, n\}$ of n jobs are to be scheduled, without interruptions, 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 non-negative integers. The precedence constraints are given by a binary relation $P \subseteq N \times N$ that describes, for certain pairs of jobs in the set, the requirement that one of the jobs must be completed before the other starts, i.e., any feasible solution must schedule job i before job j if $(i, j) \in P$. The goal is to find a feasible schedule which minimizes $\sum_{j=1}^{n} w_jC_j$, where C_j is the time at which job j completes in the schedule.

Several polynomial time 2-approximation algorithms are known [7, 8, 16, 25, 31, 32, 36], and improving on them is a major open problem (see e.g. [37]). Schulz [36], and Hall, Schulz, Shmoys & Wein [16] gave 2-approximation algorithms by using linear programming relaxations. Chudak & Hochbaum [8] gave another algorithm based on a linear programming relaxation with two variables per constraints. Independently, Pisaruk [31, 32], Chekuri & Motwani [7] and Margot, Queyranne & Wang [25], provided identical, simple 2-approximation algorithms based on Sidney's decomposition theorem [39] from 1975. Correa & Schulz [9] proved the equivalence of all these approximation algorithms.

Woeginger [43] proved that 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 not easier to approximate as the general case. On the other hand, the structure of the precedence constraints can have a big impact on the approximability of the problem. Indeed, Smith [40] showed that, in absence of precedence constraints, an optimal solution can be found by sequencing the jobs in non-increasing order of the ratio w_i/p_i. Lawler [22] gave an exact algorithm for series-parallel precedence constraints already in 1978 (other classes of polynomially solvable instances are also known and the interested reader is referred to [23] for a survey). For interval orders and convex bipartite precedence constraints, Woeginger [43] gave approximation algorithms with approximation ratio arbitrarily close to the golden ratio $\frac{1}{2}(1+\sqrt{5}) \approx 1.61803$.

In this paper we survey recent results [1, 2, 3, 4, 9] that profited from the rich dimension theory of partial orders [41]. Indeed, the set N of jobs with the precedence constraints P forms a partially ordered set (*poset*) $\mathbf{P} = (N, P)$. Dushnik & Miller [11] introduced dimension as a parameter of partial orders in 1941. The dimension is one of the most heavily studied parameters of partial orders, and many beautiful results have been obtained (see e.g. [41]).

There is a natural way to associate with a poset $\mathbf{P}$ a hypergraph $\mathbf{H_P}$, called the *hypergraph of incomparable pairs*, so that the dimension of $\mathbf{P}$ is the chromatic number of $\mathbf{H_P}$ [13]. Furthermore, the fractional dimension of $\mathbf{P}$, a generalization due to Brightwell & Scheinerman [6] is equal to the fractional chromatic number of $\mathbf{H_P}$.

Interestingly, it turns out that another classical problem in graph theory, namely the vertex cover problem, when defined on hypergraph $\mathbf{H_P}$, has a nice interpretation as a linear ordering problem. Indeed, any minimal vertex cover of $\mathbf{H_P}$ describes a total ordering complying with $\mathbf{P}$ and vice versa. In this vein, $1|prec|\sum w_jC_j$ can be seen as a special case of the weighted vertex cover on $\mathbf{H_P}$ (whereas the general version turns out to be equivalent to the weighted variant of the minimum feedback arc set problem). Moreover, in a series of papers [33, 8, 9, 1, 3] it was established that optimizing $1|prec|\sum w_jC_j$ can be reduced to computing a vertex cover in the *graph* of incomparable pairs $\mathbf{G_P}$, instead of the *hypergraph* $\mathbf{H_P}$. The (ordinary) graph $\mathbf{G_P}$ is obtained by removing from $\mathbf{H_P}$ all edges of cardinality larger than two. This result allows to apply the rich vertex cover theory to $1|prec|\sum w_jC_j$ together with the dimension theory. For example one can conclude that two-dimensional precedence constraints are solvable in polynomial time, as $\mathbf{G_P}$ is bipartite in this case [10, 13, 9], and the vertex cover problem is well-known to be solvable in polynomial time on bipartite graphs (see e.g. [19]). This considerably extends Lawler's result [22] from 1978 for series-parallel precedence constraints.

These connections between the $1|prec|\sum w_jC_j$ and the vertex cover problem on $\mathbf{G_P}$, and between dimension and coloring, yield a framework for obtaining $(2-2/f)$-approximation algorithms for classes of precedence constraints with bounded (fractional) dimension f [2, 3]. The framework is inspired by Hochbaum's approach [18] for the vertex cover problem on "easily" colorable graphs. It yields the best known approximation ratios for all previously considered special classes of precedence constraints. Furthermore, it establishes the dimension as a parameter of the complexity of an instance.

This result also triggered the first inapproximability results for $1|prec|\sum w_jC_j$. The objective function of the scheduling problem can be split into a so-called "fixed" cost and a "variable" cost (see Section 4 for details). Only the variable cost depends on the schedule, whereas the fixed cost is the same for all feasible schedules. In [4] it was shown that the variable part is as hard to approximate as the vertex cover problem. Moreover, when the fixed cost part is taken into consideration, a recent reduction from the maximum edge biclique problem [4] makes the existence of a polynomial time approximation scheme unlikely.

The structure of the paper is as follows. In Section 2 we survey dimension theory of partial orders. Section 3 covers the framework which allows to obtain better approximation algorithms for posets with bounded dimension. Section 4 is devoted to the equivalence between $1|prec|\sum w_jC_j$ and vertex cover on $\mathbf{G_P}$. Section 5 sketches the new inapproximability results for $1|prec|\sum w_jC_j$.

2 Dimension Theory of Partial Orders

A partially ordered set (or poset) formalizes the intuitive concept of an ordering, sequencing, or arrangement of the elements of a set. The theory of partially ordered sets is central to combinatorics and arises in many different contexts. We are going to introduce it by giving an example.

Consider the set $A = \{a, b, c, d\}$ and a set N consisting of subsets of A, namely $S_1 = \{a\}$, $S_2 = \{b\}$, $S_3 = \{c\}$, $S_4 = \{a, b, c\}$ and $S_5 = \{b, d\}$. The elements of the set N, called the *groundset*, can be partially ordered by the reflexive, antisymmetric and transitive relation P which we define, in the example, by using the set inclusion "$\subseteq$" (see also Figure 1): for any $S, S' \in N$ we have $(S, S') \in P$ if and only if $S \subseteq S'$. To emphasize the order concept, we write $x \leq y$ when $(x, y) \in P$.

For any $S, S' \in N$, in case $S' \leq S$ or $S \leq S'$, we will say that the pair (S, S') is *comparable*, otherwise it is called *incomparable* (for $S \neq S'$ we call S' *larger than* S, if $S \leq S'$, also denoted by $S < S'$). For instance, S_1 and S_4 are comparable since $S_1 \leq S_4$, whereas neither $S_1 \leq S_5$ nor $S_5 \leq S_1$ holds which makes the pairs (S_1, S_5) and (S_5, S_1) incomparable. The groundset N together with the partial order P define a *partially ordered set (poset)* $\mathbf{P} = (N, P)$. We denote the set of all incomparable points of a poset $\mathbf{P}$ by $inc(\mathbf{P})$. A poset that does not have any incomparable pairs is called a *linear order*.

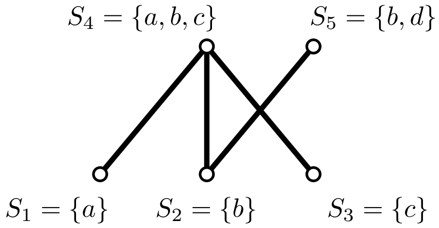

Figure 1: The Hasse diagram of the example poset $\mathbf{P}$ (each element of N is represented by a vertex on the plane and an upwards going line segment is drawn from x to y whenever $x < y$, unless there is a $z \in N$ such that $x < z < y$).

An *extension* of a poset $\mathbf{P} = (N, P)$ is any poset $\mathbf{P}' = (N, P')$ such that $P \subseteq P'$. Moreover, if P' is a linear order, we call $\mathbf{P}'$ a *linear extension* of $\mathbf{P}$. Letting aside the set-inclusion interpretation of the example poset, we can construct an extension of $\mathbf{P}$ by adding, for example, $S_3 \leq S_5$ to P. The resulting poset $\mathbf{P}'$ is depicted in Figure 2. We say that the extension $\mathbf{P}'$ *reverses* the (ordered) incomparable pair (S_5, S_3).

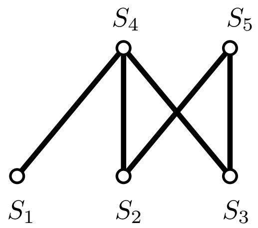

Figure 2: The Hasse diagram of the poset $\mathbf{P}'(N, P')$, an extension of the poset $\mathbf{P}(N, P)$ with $P' = P \cup \{(S_3, S_5)\}$

Constructing a linear extension that satisfies certain properties is central to several problems in combinatorics. The addressed scheduling problem $1|prec|\sum w_j C_j$ falls in this category. The construction of linear extensions of a poset also arises when determining its dimension, as explained in the following.

For a family $\mathcal{R}$ of linear extensions of P, we call $\mathcal{R}$ a *realizer* of P if $\bigcap \mathcal{R} = \mathbf{P}$, i.e. for all $a, b, \in N$, $a \leq b$ in P if and only if $a \leq b$ in every $L \in \mathcal{R}$. Note that for each incomparable pair $(a, b) \in \mathrm{inc}(\mathbf{P})$, there must be at least one linear extension in the realizer that reverses it. The least positive integer t for which there is such a realizer $\mathcal{R} = \{L_1, \ldots, L_t\}$ of $\mathbf{P}$ is called the *dimension* of $\mathbf{P}$, denoted by $\dim(\mathbf{P})$. Generalizing this concept, a *k:t*-realizer is a multiset of t linear extensions $\mathcal{R} = \{L_1, \ldots, L_t\}$ in which each incomparable pair is reversed at least k times. The *fractional dimension* of $\mathbf{P}$, denoted by $\mathrm{fdim}(\mathbf{P})$, is the infimum of the set of ratios t/k for which there exist *k:t*-realizers [6].

A natural question is for which posets one can construct a k-realizer in polynomial time. In the general case, Yannakakis [44] proved that determining whether the dimension of a poset is at most k is NP-complete for every $k \geq 3$. Moreover, Hedge & Jain [17] recently proved that it is hard to approximate the (fractional) dimension of a poset with n elements within a factor $n^{0.5-\epsilon}$, in the general case. However, for several special cases, a minimal realizer can be computed in polynomial time (see e.g. [41, 26]).

The Hypergraph of Incomparable Pairs. It is easy to see that, when constructing an extension of $\mathbf{P}$, there are some groups of incomparable pairs that cannot be reversed at the same time. Obviously, an extension of $\mathbf{P}$ cannot reverse both (S_3, S_5) and (S_5, S_3) at the same time. This implies that, unless a poset is already a linear order, any realizer needs to contain more than one linear extensions in order to reverse every incomparable pair at least once. For the pairs of incomparable pairs mentioned above, it is obvious that they cannot be reversed at the same time. There are also less obvious pairs of incomparable pairs for which this is true. By examining the Hasse-diagram of $\mathbf{P}$ (see Figure 1), one can conclude that reversing both (S_2, S_1) and (S_1, S_5) would lead to an inconsistency, i.e. a "cycle" in the ordering: adding $S_1 \leq S_2$ and using transitivity leads to $S_1 \leq S_2 \leq S_5$, which contradicts $S_5 \leq S_1$. In general, there can also be groups bigger than two pairs that cannot be all reversed at the same time without introducing contradictions (see, e.g., (S_2, S_1), (S_1, S_3) and (S_3, S_5)).

The above observations naturally lead to the definition of the *hypergraph of incomparable pairs* $\mathbf{H_P}$ of a poset $\mathbf{P}$ [13] defined as follows. The vertices of $\mathbf{H_P}$ are the incomparable pairs in $\mathbf{P}$. The edge set consists of those sets U of incomparable pairs such that:

1. No linear extension of $\mathbf{P}$ reverses all incomparable pairs in U.

2. For every proper subset of U there is a linear extension that reverses all incomparable pairs in U.

Figure 3 depicts the hypergraph of incomparable pairs for our example poset. The graph resulting from $\mathbf{H_P}$ by restricting the set E to hyperedges of size 2 (i.e. common graph edges), is called the *graph of incomparable pairs* and denoted by $\mathbf{G_P}$.

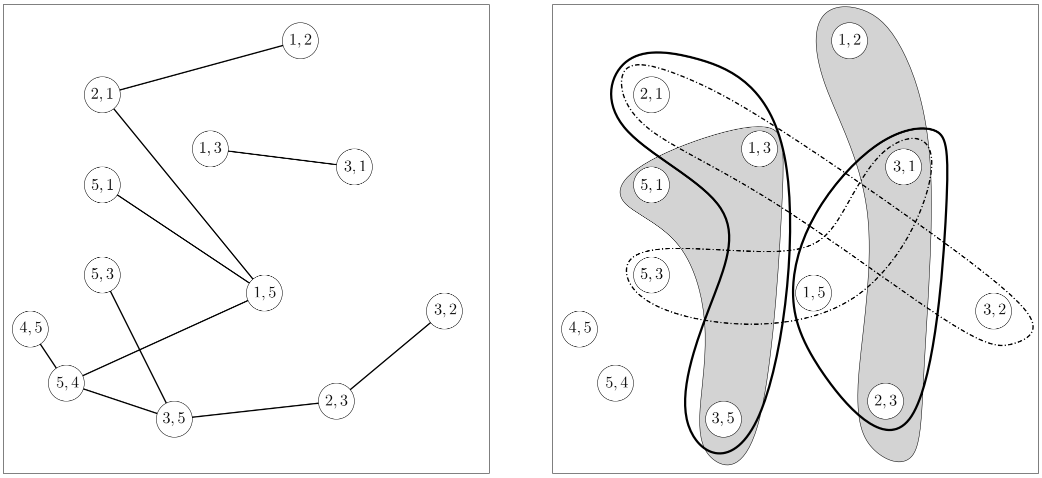

Figure 3: The graph $\mathbf{G_P}$ and the hypergraph $\mathbf{H_P}$ of incomparable pairs of our example poset $\mathbf{P}$. The *graph* of incomparable pairs $\mathbf{G_P}$ is depicted on the left. The *hypergraph* $\mathbf{H_P}$ can be obtained by adding the hyperedges on the right to the edges on the left.

Two classical problems in graph theory are the *(fractional) coloring* problem [35], and the *vertex cover* problem [19] (analogously the *hitting set* problem [19] for hypergraphs). In the fractional coloring problem, a *t:k*-coloring is an assignment of sets of k colors to vertices of a graph by using a pool of at most t colors, such that vertices of any edge receive disjoint sets. A set of vertices that covers the edges of a graph is called a *minimal vertex cover* if it does not have a proper subset that covers all the edges.

The graph theoretic problems on the graph of incomparable pairs mentioned above have interesting counterparts in combinatorics. In [13, 6] it is pointed out that the classical (fractional) coloring problem, when defined on hypergraph $\mathbf{H_P}$, is equivalent to the problem of computing the (fractional) dimension of poset $\mathbf{P}$. Moreover, it is easy to see that any linear ordering L of P "defines" a vertex cover on $\mathbf{H_P}$, i.e. the set of incomparable pairs $(x, y) \in inc(\mathbf{P})$ such that $(x, y) \in L$ forms a minimal vertex cover on $\mathbf{H_P}$. In the following, we observe that the converse is also true, namely, a minimal vertex cover problem on hypergraph $\mathbf{H_P}$ defines a linear extension of $\mathbf{P}$. Therefore, it follows that the weighted vertex cover on $\mathbf{H_P}$ defines a linear ordering problem. (Actually, $1|prec|\sum w_j C_j$ is just a special case of the linear ordering problem. It is not difficult to check that the vertex cover on $\mathbf{H_P}$ is equivalent to the weighted variant of the minimum feedback arc set problem [20, 24, 38, 21], and the independent set on $\mathbf{H_P}$ is a generalization of the maximum acyclic subgraph problem [5, 29].)

Proposition 2.1. *Let* $\mathbf{P} = (X, P)$ *be a poset which is not a linear order, and let* $\mathbf{H_P}$ *be its hypergraph of incomparable pairs. Then*

a) [13, 6] $\mathbf{H_P}$ *can be t:k-colored if and only if* $\mathbf{P}$ *has a k:t-realizer.*

b) A linear extension of $\mathbf{P}$ *defines a minimal vertex cover of* $\mathbf{H_P}$*, and a minimal vertex cover of* $\mathbf{H_P}$ *defines a linear extension of* $\mathbf{P}$.

Proof. For statement a), note that a *t:k*-coloring suggests a way of reversing each incomparable pair in at least k linear extensions, with each color corresponding to a linear extension. Since a valid coloring does not use a color common to all vertices of a hyperedge, these extensions do not contain contradictions and are therefore valid partial orders. On the other hand, a *k:t*-realizer suggests a way of coloring the vertices of the graph by assigning color i to each pair $(a, b) \in \mathrm{inc}(\mathbf{P})$ if and only if it is reversed in L_i. Since a linear extension can only reverse a proper subset of vertices for each hyperedge, there is no hyperedge that contains vertices that share a common color. Therefore, the coloring is valid.

For statement b), observe that for each $(a, b) \in \mathrm{inc}(\mathbf{P})$ a linear extension picks either $(a, b) \in P$ or $(b, a) \in P$. Since vertices of this type are connected by an edge, this vertex cover is minimal. On the other hand, assume towards contradiction that a feasible minimal vertex cover on $\mathbf{H_P}$ is not a linear extension. For this to happen, some incomparable pairs in the considered minimal vertex cover solution define a cycle $C = \{(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)\}$, with $(y_i, x_{i+1}) \in P$ for all i (modulo k). Since the pairs in C form a cycle, there must be a hyperedge $e = \{(y_1, x_1), (y_2, x_2), \dots, (y_k, x_k)\}$, and at least one of these pairs, say (y_i, x_i), is in the vertex cover. This means that both (x_i, y_i) and (y_i, x_i) are in the vertex cover and none of them can be removed, since we are assuming that the vertex cover is minimal. The latter implies the existence of two hyperedges $e_1 = \{(x_i, y_i), (a_1, b_1), \dots, (a_h, b_h)\}$ and $e_2 = \{(y_i, x_i), (c_1, d_1), \dots, (c_l, d_l)\}$ that are only covered by (x_i, y_i) and (y_i, x_i) respectively. However, note that this implies that there is a further hyperedge $e_3 = \{(a_1, b_2) \dots, (a_h, b_h), (c_1, d_1), \dots, (c_l, d_l)\}$ that remains uncovered, contradicting the feasibility of the vertex cover. □

In the following section we show how to use known results in dimension theory of partial orders to obtain approximation algorithms for two well-known ordering problems: the scheduling problem $1|prec|\sum w_j C_j$, and the maximum acyclic subgraph problem.

3 Using Dimension Theory for Approximation

The usefulness of dimension theory of partial orders for approximating ordering problems might be best seen by considering a well known graph problem, namely, the linear ordering problem (LOP) ([20, 5, 29]): Given a complete weighted directed graph, the *Linear Ordering* problem is to find a linear ordering of the vertices that maximizes the weight of the *forward* edges (edge (i, j) is a forward edge if i precedes j in the ordering). This problem is equivalent to finding a maximum acyclic subgraph of a given graph.

The linear ordering problem is NP-hard [20], and the following simple algorithm gives the best-known polynomial time computable approximation factor for the problem: according to any linear ordering of the vertices, partition the edges into two sets, those going forward in the ordering and those backward. Both sets are acyclic, and one of these sets has weight half the total weight of all the edges (the approximation ratio is therefore $1/2$).

A natural generalization of the linear ordering problem can be obtained by assuming that the set N of vertices are partially ordered to form a poset $\mathbf{P} = (N, P)$. The goal is to find a linear extension that maximizes the weight of the forward edges.[1] We observe that we cannot use the simple approximation algorithm described above for this more general version of the problem. However, the dimension of partial orders is a useful tool to obtain approximation algorithms for this more general variant. In the following we sketch this approach that will be as well the core ingredient for obtaining improved approximation algorithms for the scheduling problem $1|prec|\sum w_jC_j$ with special precedence constraints.

Let $\mathcal{R} = \{L_1, \ldots, L_t\}$ be a *k:t* realizer of $\mathbf{P}$. By definition of a *k:t* realizer, each incomparable pair is a forward edge in at least k linear extensions of $\mathcal{R}$. So if we pick uniformly at random one linear extension from $\mathcal{R}$, say L_i, the probability that an incomparable pair is a forward edge in L_i is k/t, and the expected weighted sum of the forward edges in L_i is $\frac{k}{t}W$, where W is the sum of all weights. Since W is an upper bound of the optimal value, k/t is the expected approximation ratio. We will say that a poset $\mathbf{P}$ admits an *efficiently samplable k:t*-realizer if there exists a randomized algorithm that, in polynomial time, returns any linear extension from a *k:t* realizer with probability $1/t$.

As an example, we show how to apply this framework for a well-known and heavily studied poset (see e.g. [14]), namely interval orders.

[1] Note that, whenever $(i, j) \in P$, with $i \neq j$, we can enforce that i is before j in any "good" linear ordering by choosing w_{ij} large enough. However, this would change the value of the solution together with the approximability of the problem.

An Example: Interval Orders. A poset $\mathbf{P} = (N, P)$ 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 $O(n^2)$ time [27, 30]. The fractional dimension is known to be less than 4 [6], and this bound is asymptotically tight [12]. In the following we show how to obtain a 1/4-approximation algorithm for the linear ordering problem.

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.

Theorem 3.1 (Rabinovitch [34, 14])**.** *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:t 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 $x \in A$ and $y \in B$. 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 using the described framework, we have a randomized polynomial time approximation algorithm for the linear ordering problem with interval orders, whose expected approximation ratio is 1/4. The described algorithm can easily be derandomized by using the classical method of conditional probabilities.

The Vertex Cover Problem on $\mathbf{H_P}$. Approximating the linear ordering problem where we want to *minimize* the weight of the forward edges, i.e., by Proposition 2.1 the minimum vertex cover problem on $\mathbf{H_P}$ (or, equivalently, the weighted minimum feedback arc set problem), is more delicate, as we have no longer a "good" and easy to calculate, lower bound on the optimal value. Indeed, by using the proposed framework, we can easily obtain a solution with expected value at most $(1 - k/t)W$, since the expected value of the pairs that are reversed in any uniformly picked linear extension is at least $\frac{k}{t}W$. However, to obtain a "good" approximation ratio we would need a polynomial time computable lower bound of the optimum that is "close" to W, which is not known in the general case (the approximability of the weighted minimum feedback arc set problem is a well-known open problem [42]).

As will be shown in Section 4, $1|prec|\sum w_j C_j$ can be reduced to finding a vertex cover on $\mathbf{H_P}$. The reduction preserves the approximation ratio: An α-

approximation of vertex cover of $\mathbf{H_P}$ can be turned into an α-approximation for $1|prec|\sum w_jC_j$.

However this reduction does not lead to a good approximation algorithm directly. What comes to the rescue is that instead of computing a vertex cover on $\mathbf{H_P}$, it suffices to compute a vertex cover on $\mathbf{G_P}$. This is quite remarkable as it is not the case in general that a minimal vertex cover of $\mathbf{G_P}$ corresponds to a linear extension of $\mathbf{P}$. However, as we will show in Section 4, for perturbed weights and processing times of the jobs, the minimum vertex cover instance obtained from the reduction indeed has a unique optimum solution which corresponds to a linear extension of $\mathbf{P}$ (and is a feasible vertex cover for $\mathbf{H_P}$). This property does only hold because of the special structure of the objective function of $1|prec|\sum w_jC_j$, and it is not restricted to optimal solutions. One can show that *any* vertex cover of $\mathbf{G_P}$ can be turned into one that is feasible for $\mathbf{H_P}$ without increasing its weight.

The approximation algorithm for $1|prec|\sum w_jC_j$ that makes use of this reduction is inspired by Hochbaum's preprocessing approach [18]. It works as follows. First compute an optimum solution for the linear programming relaxation of the vertex cover problem. This can be done very efficiently using a minimum cut algorithm (see [18]). It is well known that such a solution is half-integral [28]. Let V_1 be the vertices of $\mathbf{G_P}$ which attain value 1 in this solution; and let $V_{1/2}$ be the vertices which attain value $1/2$. It is not hard to see that the vertices V_1 together with a cover of the subgraph induced by $V_{1/2}$ are sufficient to cover the whole graph $\mathbf{G_P}$. To decide the cover on the induced subgraph, we apply our framework by choosing a linear extension L from the *k:t* realizer of $\mathbf{P}$ uniformly at random. By Proposition 2.1, we know that L corresponds to a vertex cover of $\mathbf{H_P}$ and therefore also of $\mathbf{G_P}$. Let C be this vertex cover. Thus, $V_1 \cup (V_{1/2} \cap C)$ is a vertex cover for $\mathbf{G_P}$ and in expectation it is a $(2 - \frac{2}{t/k})$-approximation of the minimum vertex cover. As mentioned above, this solution can then be turned into a solution for $1|prec|\sum w_jC_j$with the same approximation ratio [1].

In Table 1 we list some well-known posets that admit an efficiently samplable *k:t*-realizer with the corresponding approximation ratios that can be achieved for LOP and $1|prec|\sum w_jC_j$. For all of them it is easy to obtain a derandomized approximation algorithm. The interested reader can find more details in [2, 3].

4 The Vertex Cover of $\mathbf{G_P}$ and Scheduling with Precedence Constraints

In this section we show that the scheduling problem can be reduced to the vertex cover problem on $\mathbf{G_P}$. We start by reducing the scheduling problem to the weighted vertex cover problem on the *hypergraph* $\mathbf{H_P}$.

Table 1: Approximation Ratios

Poset	**LOP**	$1\|prec\|\sum w_jC_j$
2-dimensional	1/2	1
semi-orders	1/3	4/3
convex bipartite	1/3	4/3
interval-orders	1/4	3/2
interval dimension 2	1/8	7/4
Bounded degree d	$1/(d+1)$	$2-\frac{2}{d+1}$

Problem $1|prec|\sum w_jC_j$ and the vertex cover on $\mathbf{H_P}$. Let us consider an instance of the scheduling problem where the precedence constraints are given by poset $\mathbf{P} = (N, P)$. The feasible solutions of problem $1|prec|\sum w_jC_j$ can be expressed in terms of sets $E \subseteq \text{inc}(\mathbf{P})$ such that $P \cup E$ forms a linear extension of $\mathbf{P}$. It is not hard to see that the objective function value of solution E can be written as follows.

$$\sum_{i\in N} w_iC_i = \underbrace{\sum_{(i,j)\in E} p_iw_j}_{\text{variable part}} + \underbrace{\sum_{(i,j)\in P} p_iw_j}_{\text{fixed part}} . \tag{1}$$

The first summand is called *variable part* of the objective function, since its value depends on the set E of incomparable pairs that we choose to be part of the linear extension of $\mathbf{P}$. The second summand is called *fixed part*, as its value does not depend on E, i.e. it is contained in all feasible solutions. Finding an optimal solution for $1|prec|\sum w_jC_j$ can therefore be reduced to finding a optimal solution for the variable part of the objective function. By Proposition 2.1, it is easy to check that the variable part of the problem is equivalent to the vertex cover on $\mathbf{H_P}$ where the the weight of any $(i, j) \in inc(\mathbf{P})$ is equal to p_iw_j.

Problem $1|prec|\sum w_jC_j$ and the vertex cover on $\mathbf{G_P}$. For any given scheduling instance S of $1|prec|\sum w_jC_j$, where precedence constraints are given by a poset $\mathbf{P} = (N, P)$, we will use $G^S_{\mathbf{P}}$ to denote the graph of incomparable pairs $\mathbf{G_P}$ where any vertex corresponding to $(i, j) \in inc(\mathbf{P})$ has weight equal to p_iw_j. In what follows, the Vertex Cover problem on $G^S_{\mathbf{P}}$ will be denoted by π. For problem π, the feasibility conditions on E are slightly weaker than for $1|prec|\sum w_jC_j$. We give a detailed description in the following.

Recall that in $G^S_{\mathbf{P}}$ there is an edge between two vertices (i, j) and (k, l) of $G^S_{\mathbf{P}}$ if and only if adding the arcs (j, i) and (l, k) would lead to a cycle in $\mathbf{P}$. This translates into three conditions which E has to fulfill, depicted in Figure 4. Condition (i) states that between any pair of unordered jobs, at least one of (i, j) and (j, i) must be in E. Condition (ii) states that for any triple of jobs i, j, k, with $(i, j) \in inc(\mathbf{P})$, $(j, k) \in inc(\mathbf{P})$ and $(i, k) \in P$, at least one of (i, j) and (j, k) must be in E. Finally, Condition (iii) requires that for a quadruple with $(i, j) \in inc(\mathbf{P})$, $(k, j) \in P$, $(k, l) \in inc(\mathbf{P})$ and $(i, l) \in P$, at least one of (i, j) and (k, l) must be in E.

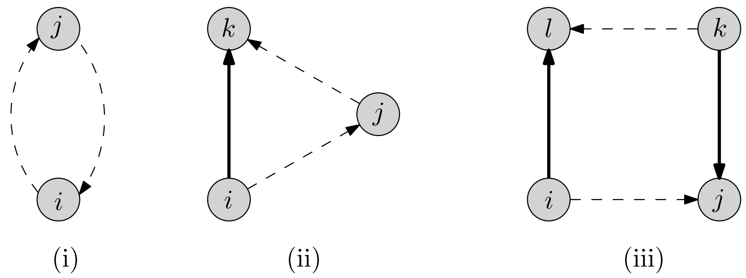

Figure 4: In each of the three cases, at least one of the two dotted arcs must be present in E.

The objective function of π translates to $\sum_{(i,j)\in E} p_i w_j$, which is exactly the variable part of the objective function of $1|prec|\sum w_j C_j$.

It turns out that the vertex cover on $G^S_{\mathbf{P}}$ is a proper formulation of the variable part of the scheduling problem.

Theorem 4.1 ([1, 9, 8])**.** *For any given instance S of $1|prec|\sum w_j C_j$, any vertex cover of $G^S_{\mathbf{P}}$ can be turned into a feasible solution for S in polynomial time, without deteriorating the variable part of the objective value.*

For the sake of simplicity, we will only give the proof of Theorem 4.1 for an optimal vertex cover of $G^S_{\mathbf{P}}$. For a complete proof refer to [1].

Again for simplicity let us perturb the weights and the processing times of the jobs by replacing their original weight w_i and processing time p_i by $w_i + \varepsilon^i$ and $p_i + \varepsilon^i$, respectively. For $\varepsilon > 0$ small enough, the optimal solution with the perturbed values will also be optimal for the original values.

Lemma 4.2. *[9] An optimum solution E for π has either $(i, j) \in E$ or $(j, i) \in E$, but not both.*

Proof Sketch. Removing arcs from E does always decrease the objective value as all the weights and processing times are positive after perturbation.

We now prove that if neither (i, j) nor (j, i) can be removed from E, then E is infeasible itself. To see this, assume both (i, j) and (j, i) could not be removed because each of them was blocked by one of the two structures from Conditions *(ii)* and (*iii*). There are nine cases to be considered. Since all of them are very similar in nature, only the one depicted in Figure 5 is considered here.

In the figure, (i, j) cannot be removed because of Condition (*ii*) as $(j, k) \notin E$ and (j, i) cannot be removed because of Condition (*iii*) as $(m, l) \notin E$. Note that transitivity of P implies $(m, k) \in P$. Hence, Condition (*iii*) applied to the quadruple m, k, j, l is violated. □

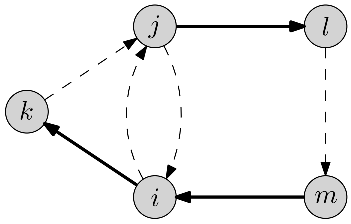

Figure 5: Contradiction to the assumption that removing either (i, j) or (j, i) was not possible.

Lemma 4.3. *There exists an optimal solution for π which is also feasible for $1|prec|\sum w_j C_j$.*

Proof. Condition (*i*) together with Lemma 4.2 implies that any optimal solution strictly orders every pair of vertices. We still have to prove that there are no oriented cycles in $E \cup P$. Since all pairs of vertices are strictly ordered, it is enough to prove that there are no oriented 3-cycles in $E \cup P$. 3-cycles involving precedence constraints are already dealt with by Condition (*ii*). The only 3-cycles we still have to consider are those involving three arcs from E. We denote a 3-cycle involving the arcs (i, j), (j, k), and (k, i) by $\langle ijk \rangle$.

Let E be an optimum solution for π which contains at least one 3-cycle. We will show that this solution can still be improved, thus contradicting optimality.

For $q \in N$, let E_q be the set of arcs (i, j) for which $\langle ijq \rangle \subseteq E$. Figure 6 depicts, up to symmetry, the cases in which the reversal of (i, j) alone would violate one of the Conditions (*ii*) and (*iii*). The nice property of E_q is that reversing all arcs of E_q together yields another feasible solution. Indeed, observe that the reversal of an arc (i, j) cannot lead to an infeasible solution. This is because any arc of E which could block the reversal by violating Condition (*ii*) or (*iii*) gets reversed as well. In Figure 6, the reversal of (i, j) is apparently blocked by (i, k), (k, j), and (l, k), respectively. Note that all curved arcs are implied by the straight arcs and the precedence constraints: if they were pointing in the other direction, Condition (*ii*) would be violated. But the presence of the curved arcs allow to conclude that the blocking arcs are in E_q as well since they form a 3-cycle together with q. Hence, (i, j) can be reversed in all cases.

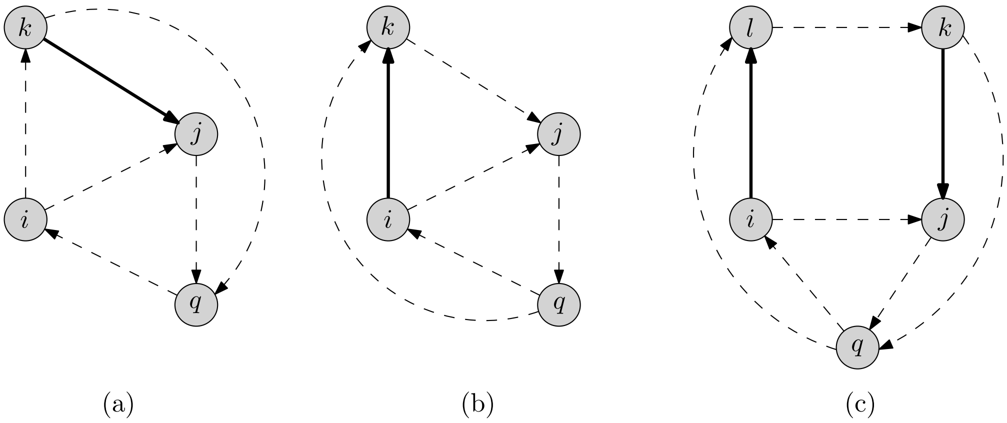

Figure 6: Arcs blocking the reversal of (i, j) are in E_q as well.

Denote by Δ_q the change in the objective function after reversing the arcs in E_q. It holds

$$\Delta_q = \sum_{\langle ijq\rangle \subseteq E} p_j w_i - p_i w_j.$$

Because of the perturbation, it holds that $\Delta_q \neq 0$ unless $E_q = \emptyset$. It remains to show that there always exists a q with $\Delta_q < 0$. Observe that

$$\begin{aligned}
\sum_{q\in N} p_q \Delta_q &= \sum_{q\in N} p_q \sum_{\langle ijq\rangle \subseteq E} \left(p_j w_i - p_i w_j\right) \\
&= \sum_{q\in N} p_q \sum_{\langle ijq\rangle \subseteq E} p_j w_i - \sum_{q\in N} p_q \sum_{\langle ijq\rangle \subseteq E} p_i w_j \\
&= \sum_{\langle ijk\rangle \subseteq E} \left(p_k p_j w_i + p_i p_k w_j + p_j p_i w_k\right) \\
&\quad - \sum_{\langle ijk\rangle \subseteq E} \left(p_k p_i w_j + p_i p_j w_k + p_j p_k w_i\right) \\
&= 0.
\end{aligned}$$

Therefore, since $E_q \neq \emptyset$ for some q and we have $p_k > 0$ for all $k \in N$ after perturbation, there must exist a q with $\Delta_q < 0$. $\square$

5 On the Hardness of the Scheduling Problem

In the previous sections, we obtained the best known approximation algorithms for several classes of precedence constraints, by exploiting the vertex cover nature of the problem, and deriving better than 2-approximation algorithms for the variable part. It is a natural and interesting question to understand if a better than 2-approximate solution for the general version of the problem can be obtained in a similar vein. In the following subsection we show this to be unlikely by showing that the variable part of $1|prec|\sum w_jC_j$ is in fact equivalent to vertex cover in terms of approximability.

In the second part of this section, we establish a nice relationship between the maximum edge biclique problem and the scheduling problem $1|prec|\sum w_jC_j$. This relationship rules out, under some fairly standard assumption, the existence of a PTAS for $1|prec|\sum w_jC_j$ and also explains the difficulty in proving inapproximability results for $1|prec|\sum w_jC_j$, since providing strong inapproximability results for the maximum edge biclique problem is a major open problem in complexity theory.

5.1 Hardness of Variable Part

We show that approximating the variable cost of $1|prec|\sum w_jC_j$ is equivalent to approximating the vertex cover problem. Theorem 4.1 implies that minimizing the variable cost of $1|prec|\sum w_jC_j$ 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 solving the variable part is essentially equal to finding a minimum vertex cover of G.

Theorem 5.1. *Approximating the variable cost of* $1|prec|\sum w_jC_j$ *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 $r \geq 1$ and choose $k > n^2r/\epsilon$. 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 $v_i' \to v_j''$ and $v_j' \to v_i''$. Finally, we add $v_i' \to v_j''$ for every i, j with $i < j$. See Figure 7 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

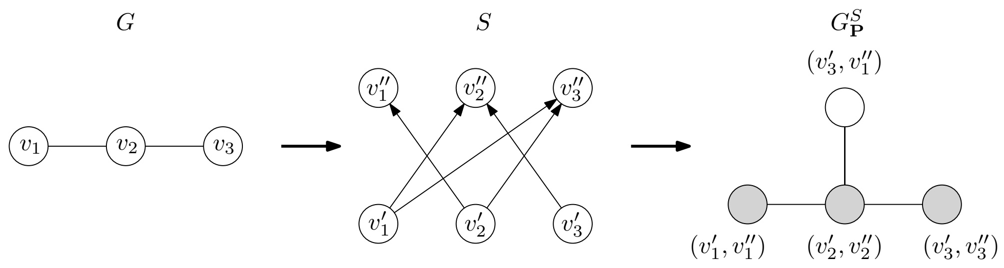

Figure 7: The transformation of a graph G.

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 $(v_i, v_j) \in G$.

Using the connection between S and $G^S_{\mathbf{P}}$ provided by Theorem 4.1 and the close relation between $G^S_{\mathbf{P}}$ and G, it is easy to see that an r-approximation algorithm for the optimum variable cost of $1|prec|\sum w_jC_j$ would imply an approximation algorithm for vertex cover with approximation ratio $r(1 + n^2/k) < (r + \epsilon)$. $\square$

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

5.2 Ruling out a PTAS

We show a nice relationship between the maximum edge biclique problem (MEB) and $1|prec|\sum w_jC_j$. This relationship together with the currently best inapproximability result for MEB yields that the scheduling problem has no PTAS unless NP-complete problems can be solved in randomized subexponential time.

Definition 5.2. *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.

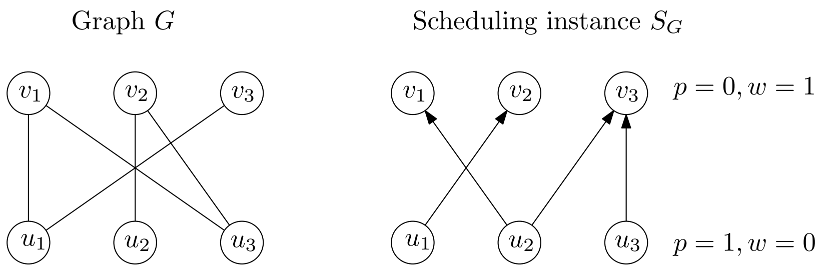

Figure 8: An example of a graph G with its associated scheduling instance S_G.

The intuition behind the relationship between $1|prec|\sum w_j C_j$ and MEB is best seen by considering the 2D Gantt chart. 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. 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 9), or equivalently, the area above the work line subtracted from n^2. The relationship to MEB now follows from the following subtle observation: each point (s, t) on the work line of a schedule of S_G defines an edge biclique of G of size at least $(n - s)t$, 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 t of them), see striped area in Figure 9. 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.

Formalizing the above intuition gives us

Lemma 5.3. *Let $val(\sigma^*)$ 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*

$$n^2 - an^2(\ln 1/a + 2) \leq val(\sigma^*) \leq n^2 - an^2.$$

Proof. We start by showing that $\text{val}(\sigma^*) \leq n^2 - an^2$. Let $A \subseteq U, B \subseteq V$ be an edge biclique solution with value $|A| \cdot |B| = an^2$. Consider a schedule σ that schedules the jobs in the order $U \setminus A \to B \to A \to V \setminus B$. The feasibility of such a schedule can be seen by observing that there is no precedence constraints from the jobs in

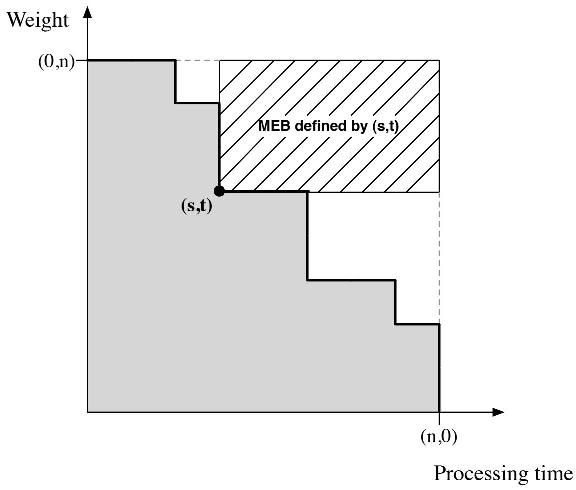

Figure 9: 2D Gantt chart representation of a schedule.

A to the jobs in B. The bound now follows since $\mathrm{val}(\sigma^*) \leq \mathrm{val}(\sigma)$ and

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

To prove the lower bound $n^2 - an^2(\ln 1/a + 2) \leq val(\sigma^*)$ 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 has not been scheduled, say A, has no precedence constraints to the set of jobs of V that has 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

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

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

□

A well-known way to boost hardness results for the Maximum Clique problem is the usage of so–called graph products. The same technique can be applied to the Maximum Edge Biclique problem and the result in [4] can be stated as follows.

Theorem 5.4 ([4]). *There exist positive constants b and ϵ so that for all constants $u > 0$, there is no polynomial algorithm that decides whether an n by n bipartite graph*

- *(Completeness:) has an edge biclique of value $(b + \epsilon)^u n^2$ or*
- *(Soundness:) has no edge biclique of value $b^u n^2$,*

unless NP-complete problems can be solved in randomized subexponential time.

Combining the above theorem with the bounds of Lemma 5.3, we have that, in the completeness case, S_G has a schedule of value $n^2 (1 - (b + \epsilon)^u)$ whereas, in the soundness case, all schedules of S_G have value at least $n^2 (1 - b^u(\ln 1/b^u + 2))$. Selecting u to be a sufficiently large constant (depending on b and ϵ) yields

Theorem 5.5. *Problem $1|prec|\sum w_j C_j$ has no PTAS unless NP-complete problems can be solved in randomized subexponential time.*

Acknowledgements

This research is supported by Swiss National Science Foundation project 200021-104017/1, "Power Aware Computing", by the Swiss National Science Foundation project 200020-109854, "Approximation Algorithms for Machine scheduling Through Theory and Experiments II", and by the Swiss National Science Foundation project PBTI2-120966, "Scheduling with Precedence Constraints". The first author is supported by Nuffield Foundation Grant NAL32608. The authors would like to thank Gerhard Woeginger for the invitation to write this survey.

References

[1] C. Ambühl and M. Mastrolilli. Single machine precedence constrained scheduling is a vertex cover problem. In *Proceedings of the 14th Annual European Symposium on Algorithms (ESA)*, volume 4168 of *Lecture Notes in Computer Science*, pages 28–39. Springer, 2006.

[2] C. Ambühl, M. Mastrolilli, N. Mutsanas, and O. Svensson. Scheduling with precedence constraints of low fractional dimension. In *Proceedings of IPCO 2007*, volume LNCS 4168, pages 28–39. Springer, 2007.

[3] C. Ambühl, M. Mastrolilli, and O. Svensson. Approximating precedence-constrained single machine scheduling by coloring. In *Proceedings of APPROX + RANDOM*, volume 4110 of *Lecture Notes in Computer Science*, pages 15–26. Springer, 2006.

[4] C. Ambühl, M. Mastrolilli, and O. Svensson. Inapproximability results for sparsest cut, optimal linear arrangement, and precedence constraint scheduling. In *Proceedings of FOCS 2007*, 2007.

[5] S. Arora, A. M. Frieze, and H. Kaplan. A new rounding procedure for the assignment problem with applications to dense graph arrangement problems. In *FOCS*, pages 21–30, 1996.

[6] G. R. Brightwell and E. R. Scheinerman. Fractional dimension of partial orders. *Order*, 9:139–158, 1992.

[7] C. Chekuri and R. Motwani. Precedence constrained scheduling to minimize sum of weighted completion times on a single machine. *Discrete Applied Mathematics*, 98(1-2):29–38, 1999.

[8] F. A. Chudak and D. S. Hochbaum. A half-integral linear programming relaxation for scheduling precedence-constrained jobs on a single machine. *Operations Research Letters*, 25:199–204, 1999.

[9] J. R. Correa and A. S. Schulz. Single machine scheduling with precedence constraints. *Mathematics of Operations Research*, 30(4):1005–1021, 2005. Extended abstract in Proceedings of the 10th Conference on Integer Programming and Combinatorial Optimization (IPCO 2004), pages 283–297.

[10] J. Doignon, A. Ducamp, and J. Falmagne. On realizable biorders and the biorder dimension of a relation. *Journal of Mathematical Psychology*, 28:73–109, 1984.

[11] B. Dushnik and E. Miller. Partially ordered sets. *American Journal of Mathematics*, 63:600–610, 1941.

[12] S. Felsner and W. T. Trotter. On the fractional dimension of partially ordered sets. *DMATH: Discrete Mathematics*, 136:101–117, 1994.

[13] S. Felsner and W. T. Trotter. Dimension, graph and hypergraph coloring. *Order*, 17(2):167–177, 2000.

[14] P. C. Fishburn. *Interval Orders and Interval Graphs*. John Wiley and Sons, 1985.

[15] R. Graham, E. Lawler, J. K. Lenstra, and A. H. G. Rinnooy Kan. Optimization and approximation in deterministic sequencing and scheduling: A survey. In *Annals of Discrete Mathematics*, volume 5, pages 287–326. North–Holland, 1979.

[16] L. A. Hall, A. S. Schulz, D. B. Shmoys, and J. Wein. Scheduling to minimize average completion time: off-line and on-line algorithms. *Mathematics of Operations Research*, 22:513–544, 1997.

[17] R. Hegde and K. Jain. The hardness of approximating poset dimension. *Electronic Notes in Discrete Mathematics (ENDM)*, 29:435 – 443, 2007.

[18] D. S. Hochbaum. Efficient bounds for the stable set, vertex cover and set packing problems. *Discrete Applied Mathematics*, 6:243–254, 1983.

[19] D. S. Hochbaum, editor. *Approximation Algorithms for NP-hard Problems*. PWS Publishing Company, 1997.

[20] R. M. Karp. Reducibility among combinatorial problems. In *Complexity of Computer Computations*. Plenum Press, 1972.

[21] C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *STOC*, pages 95–103, 2007.

[22] E. L. Lawler. Sequencing jobs to minimize total weighted completion time subject to precedence constraints. *Annals of Discrete Mathematics*, 2:75–90, 1978.

[23] E. L. Lawler, J. K. Lenstra, A. H. G. R. Kan, and D. B. Shmoys. Sequencing and scheduling: Algorithms and complexity. In S. C. Graves, A. H. G. R. Kan, and P. Zipkin, editors, *Handbooks in Operations Research and Management Science*, volume 4, pages 445–552. North-Holland, 1993.

[24] F. T. Leighton and S. Rao. An approximate max-flow min-cut theorem for uniform multicommodity flow problems with applications to approximation algorithms. In *FOCS*, pages 422–431, 1988.

[25] F. Margot, M. Queyranne, and Y. Wang. Decompositions, network flows and a precedence constrained single machine scheduling problem. *Operations Research*, 51(6):981–992, 2003.

[26] R. H. Möhring. Computationally tractable classes of ordered sets. *Algorithms and Order*, pages 105–194, 1989.

[27] R. H. Möhring. Computationally tractable classes of ordered sets. In I. Rival, editor, *Algorithms and Order*, pages 105–193. Kluwer Academic, 1989.

[28] G. L. Nemhauser and L. E. Trotter. Vertex packings: Structural properties and algorithms. *Mathematical Programming*, 8:232–248, 1975.

[29] A. Newman and S. Vempala. Fences are futile: On relaxations for the linear ordering problem. In *IPCO*, pages 333–347, 2001.

[30] C. H. Papadimitriou and M. Yannakakis. Scheduling interval-ordered tasks. *SIAM Journal of Computing*, 8:405–409, 1979.

[31] N. Pisaruk. The boundaries of submodular functions. *J. Comput. Math. Math. Phys.*, 32:1769–1783, 1992.

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

[33] C. N. Potts. An algorithm for the single machine sequencing problem with precedence constraints. *Mathematical Programming Study*, 13:78–87, 1980.

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

[35] E. R. Scheinerman and D. H. Ullman. *Fractional Graph Theory*. John Wiley and Sons Inc., 1997.

[36] A. S. Schulz. Scheduling to minimize total weighted completion time: Performance guarantees of LP-based heuristics and lower bounds. In *Proceedings of the 5th Conference on Integer Programming and Combinatorial Optimization (IPCO)*, pages 301–315, 1996.

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

[38] P. D. Seymour. Packing directed circuits fractionally. *Combinatorica*, 15(2):281–288, 1995.

[39] J. B. Sidney. Decomposition algorithms for single-machine sequencing with precedence relations and deferral costs. *Operations Research*, 23:283–298, 1975.

[40] W. E. Smith. Various optimizers for single-stage production. *Naval Research Logistics Quarterly*, 3:59–66, 1956.

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

[42] V. V. Vazirani. *Approximation Algorithms*. Springer, 2001.

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

[44] M. Yannakakis. On the complexity of partial order dimension problem. *SIAM Journal on Algebraic and Discrete Methods*, 22(3):351–358, 1982.