Partially-Ordered Knapsack
and Applications to Scheduling

Stavros G. Kolliopoulos1⋆ and George Steiner2⋆⋆

1 Department of Computing and Software, McMaster University
stavros@mcmaster.ca

2 Management Science and Information Systems, McMaster University
steiner@mcmaster.ca

Abstract. In the partially-ordered knapsack problem (POK) we are given a set \( N \) of items and a partial order \( \prec_P \) on \( N \). Each item has a size and an associated weight. The objective is to pack a set \( N' \subseteq N \) of maximum weight in a knapsack of bounded size. \( N' \) should be precedence-closed, i.e., be a valid prefix of \( \prec_P \). POK is a natural generalization, for which very little is known, of the classical Knapsack problem. In this paper we advance the state-of-the-art for the problem through both positive and negative results. We give an FPTAS for the important case of a 2-dimensional partial order, a class of partial orders which is a substantial generalization of the series-parallel class, and we identify the first non-trivial special case for which a polynomial-time algorithm exists. We also characterize cases where the natural linear relaxation for POK is useful for approximation and we demonstrate its limitations. Our results have implications for approximation algorithms for scheduling precedence-constrained jobs on a single machine to minimize the sum of weighted completion times, a problem closely related to POK.

1 Introduction

Let a partially-ordered set (poset) be denoted as \( P = (N, \prec_P) \), where \( N = \{1, 2, ..., n\} \). A subset \( I \subseteq N \) is an order ideal (or ideal or prefix) of \( P \) if \( b \in I \) and \( a \prec_P b \) imply \( a \in I \). In the partially-ordered knapsack problem (denoted POK), the input is a tuple \( P = (N, \prec_P, w, p, h, l) \) where \( P \) is a poset, \( w: N \to R^+ \), \( p: N \to R^+ \), and \( h, l \) are scalars in \((1, +\infty)\). For a set \( S \subseteq N \), \( p(S) \) \( (w(S)) \) denotes \( \sum_{i \in S} p_i \) \( (\sum_{i \in S} w_i) \). We can think of being given a knapsack of capacity \( p(N)/l \). The sought output is an ideal \( N' \) that fits in the knapsack, i.e., \( p(N') \leq p(N)/l \), and for which \( w(N') \geq w(N)/h \). In the optimization version of POK only the knapsack capacity \( p(N)/l \) is given and one seeks to maximize \( w(N') \). The two formulations are polynomial-time equivalent from the algorithmic perspective. In our exposition we adopt each time the one which is more convenient. A \( \rho \)-approximation algorithm, \( \rho < 1 \), finds an ideal \( N' \) such

⋆ Research partially supported by NSERC Grant 227809-00

⋆⋆ Research partially supported by NSERC Grant OG0001798
that \( p(N') \leq p(N)/l \) and \( w(N') \) is at least \( \rho \) times the optimum. For simplicity we shall sometimes denote a POK instance as a triple \( (N, h, l) \) with \( \prec_p, w, p \) implied from the context. Similarly, we occasionally abuse notation and denote a poset by \( N \), or omit \( P \) from \( \prec_p \) when this leads to no ambiguity.

POK is a natural generalization of the Knapsack problem. An instance of the latter is a POK instance with an empty partial order. Johnson and Niemi \[13\] view POK as modeling, e.g., an investment situation where every investment has a cost and a potential profit and in which certain investments can be made only if others have been made previously. POK is strongly NP-complete, even when \( p_i = w_i, \forall i \in N \), and the partial order is bipartite \[13\] and hence does not have an FPTAS, unless \( P = NP \). To our knowledge, this is the only hardness of approximation result known. However, the problem is believed to have much more intricate structure. There is an approximation-preserving reduction from the densest \( k \)-subgraph problem (\( DkS \)) to POK. No NP-hardness of approximation result exists for \( DkS \) but the best approximation ratio currently known is \( O(n^\delta) \), \( \delta < 1/3 \) \[8\]. Very recently, Feige showed that \( DkS \) is hard to approximate within some constant factor under an assumption about the average-case complexity of 3SAT \[7\]. In terms of positive results for POK, there is very little known. In 1983 Johnson and Niemi gave an FPTAS for the case when the precedence graph is a directed out-tree \[13\]. Recently there has been revived interest due to the relevance of POK for scheduling. It is known that an \( O(1) \)-factor approximation for a special type of POK instances leads to a \( (2 - \delta) \)-approximation for minimizing average completion time of precedence-constrained jobs on a single machine, a problem denoted as \( 1|\text{prec}|\sum w_j C_j \). Improving on the known factor of 2 for the latter problem (see, e.g., \[3\], \[4\], \[10\], \[15\]) is one of the major open questions in scheduling theory \[18\]. The relationship between POK and \( 1|\text{prec}|\sum w_j C_j \) was explored in a recent paper by Woeginger \[21\].

Due to the scheduling connection, we adopt scheduling terminology for POK instances: items in \( N \) are jobs, function \( w \) assigns weights and function \( p \) processing time. An instance \( (N, h, l) \) is a weight-majority instance if \( h < l \), and time-majority if \( h > l \). To our knowledge, Woeginger gave after many years the first new results for POK by showing pseudopolynomial algorithms for the cases where the underlying partial order is an interval order or a bipartite convex order \[21\]. We use the notion of convex orders ourselves so we proceed to define it. The comparability graph of a bipartite poset \( (X, Y; \prec) \) is a bipartite graph \( G = (X, Y; E) \) and by convention, the set of maximal elements of the partial order is \( Y \). A bipartite poset is bipartite convex if its comparability graph is convex: the vertices can be ordered so that the set of predecessors of any job from \( Y \) forms an interval within the jobs in \( X \).

Our contribution. In this paper we advance the state-of-the-art for POK through both positive and negative results. Our positive results are based on structural information of posets which has not been exploited before in approximation algorithms. One of the main applications of posets is in scheduling problems but there are only a few relevant results (e.g., \[5\], \[10\], ). Moreover, these results are usually derived either by simple greedy scheduling or by relying on an LP solution to
resolve the ordering. However, a large amount of combinatorial theory exists for posets. Tapping this source can only help in designing approximation algorithms. Following this approach, we obtain combinatorial algorithms for comprehensive classes of POK instances. These lead to improved approximation algorithms for the corresponding cases of $1\text{prec}\sum w_jC_j$. Perhaps ironically, we then show that the natural LP relaxation for POK provides only limited information.

The first part of our paper deals with the complexity of POK on two comprehensive classes of partial orders. First, we give an FPTAS for POK when the underlying order is 2-dimensional. Second, we give a polynomial-time algorithm for a special class of bipartite orders.

2-dimensional orders. In Section 2 we provide a bicriteria FPTAS for POK when the underlying order is 2-dimensional. It achieves a simultaneous $(1 + \varepsilon)$-approximation for weight and processing time. We proceed to give background on 2-dimensional orders. A linear extension of a poset $P = (N, \prec_P)$ is a linear (total) order $L$ with $a \prec_P b$ implying $a \prec_L b$ for $a, b \in N$. Every poset $P$ can be defined as the intersection of its linear extensions (as binary relations) [20]. The minimum number of linear extensions defining $P$ in this way is the dimension of $P$, denoted by $\dim P$. It is well known that $\dim P = 2$ exactly when $P$ can be embedded into the Euclidean plane so that $a \prec_P b$ for $a, b \in N$ if and only if the point corresponding to $a$ is not to the right and not above the point corresponding to $b$. 2-dimensional posets were first characterized by Dushnik and Miller [6] and they can be recognized and their two defining linear extensions can be found in polynomial time. However, recognizing whether $\dim P = k$ for any $k \geq 3$ is NP-complete [22]. POK is NP-complete on 2-dimensional partial orders, since the empty partial order is also of dimension 2. It is well known that every directed out-tree poset is series-parallel and that every series-parallel poset is also of dimension 2, but the class of 2-dimensional posets is substantially larger. For example, while the class of series-parallel posets can be characterized by a single forbidden subposet, posets of dimension 2 cannot be defined by a finite list of forbidden substructures. Thus our FPTAS for 2-dimensional POK represents a substantial addition to previously known positive results on directed out-trees [13] and other classes [21]. For a review of the extensive literature on 2-dimensional posets, we refer the reader to [16].

Complement of chordal bipartite orders. A POK instance is called Red-Blue if $\forall a \in N$, either $w_a = 0$ (a is red) or $p_a = 0$ (a is blue). Red-Blue bipartite instances of POK are of particular interest, because any $O(1)$-approximation to these would yield improved approximation results for $1\text{prec}\sum w_jC_j$ [3, 21]. Note also that solving any POK instance can be reduced in an approximation-preserving manner to solving a Red-Blue bipartite instance (cf. Sec. 6). In Section 3 we give a polynomial-time algorithm for POK on Red-Blue bipartite instances where the comparability graph has the following property: its bipartite complement is chordal bipartite. Chordal bipartite graphs are bipartite graphs in which every cycle of length 6 or more has a chord. They form a large class of perfect graphs, containing, for example, convex and biconvex bipartite graphs,
bipartite permutation graphs (the comparability graphs of bipartite posets of
dimension 2), bipartite distance hereditary graphs and interval bigraphs and
they can be recognized in polynomial time [17]. For an excellent overview of
these graph classes, the reader is referred to [1]. To the best of our knowledge,
our result identifies the first nontrivial special case of POK which is solvable
in polynomial time. All other solvable cases, e.g., when the poset is a rooted
tree [13], an interval order [21], a convex bipartite poset [21] or a series-parallel
poset, include the case when the partial order is empty, i.e., the classical Knapsack
problem. Hence their best algorithm can only be pseudopolynomial, unless
\( P = NP \). In contrast, the class we define does not include Knapsack; moreover
it is the “maximal” possible in \( P \) since without the Red-Blue constraint the
problem becomes NP-hard even on these restricted posets. We also give an FP-
TAS for POK with general \( w \) and \( p \) functions and comparability graph whose
bipartite complement is chordal bipartite.

As a corollary to our POK results, we obtain in Section 4 an 1.61803-
approximation for \( 1|\text{prec}| \sum w_j C_j \) when the partial order of the jobs falls in one
of the two classes we described above. Our derivation uses machinery developed
by Woeginger in [21].

In the second part of our paper we turn our attention to the problem with
a general partial order. We study the natural linear relaxation and provide ins-
sights on the structure of optimal solutions. On the positive side we provide
in Section 5 a bicriteria-type approximation for weight and processing time on
weight-majority instances. On the negative side, we show that the LP solution is inadequate in the general case. Let the rank of an ideal \( I \) be defined as
\( w(I)/p(I) \). Poset \( N \) is indecomposable if its maximum rank ideal is the entire set. It is well-known that in order to improve on the 2-approximation for
\( 1|\text{prec}| \sum w_j C_j \) one needs only to consider indecomposable instances [3, 19]. As
part of our contribution, we show that indecomposability affects POK, although
in a different manner. In Section 6 we show that if the input is indecomposable
the LP-relaxation provides essentially no information since the optimum solution
assigns the same value to all the variables (cf. Theorem 8). We base our analysis
on the fact that the dual LP is related to a transportation problem. We also
exploit the flow connection to show non-constructively an upper bound on the
integrality gap of the LP-relaxation that holds regardless of indecomposability.

Our guarantee for the weight in our bicriteria-type result is not an approxi-
mation ratio in the classical sense (cf. Theorem 7) and as said applies only for
the weight-majority case. We give evidence in Section 7 that both these limi-
tations of our algorithm reflect rather the difficulty of the problem itself.
We show that an \( O(1) \)-approximation algorithm for the POK problem restricted to
weight-majority instances, would imply a breakthrough \( (2-\delta) \)-approximation for
\( 1|\text{prec}| \sum w_j C_j \). This is a rather surprising fact given that any weight-majority
instance defined on an indecomposable poset is by definition infeasible. As men-
tioned, the hard case for \( 1|\text{prec}| \sum w_j C_j \) is precisely the one where the underlying
poset is indecomposable.
2 POK on 2-Dimensional Orders

Consider a POK instance \((P = (N, \prec_P), b, l)\) where \(\prec_P\) is 2-dimensional. For simplicity we denote the knapsack capacity \(p(N)/l\) by \(b\) and \(w(N)\) by \(W\). Without loss of generality we assume that processing times, weights and the knapsack capacity are all integers. Define the \(b \times W\) feasibility matrix \(M(N, b, W)\) as follows: For each integer \(p \in [1, b]\) and \(w \in [1, W]\), the corresponding element \(m_{p,w} = 1\) if there is an ideal \(I\) such that \(p(I) = p \leq b\) and \(w(I) = w\); otherwise \(m_{p,w} = 0\). It is clear that any POK problem can be solved by computing its matrix \(M(N, b, W)\) and then finding \(\max\{w \mid m_{p,w} = 1\}\). For any \(S \subseteq N\), define \(M(S, b, w(S))\) as the feasibility matrix for the induced POK problem on \(S\). For \(k \in S\) define further \(M'(S, k, b, w(S))\) as the feasibility matrix for the induced POK problem on \(S\) which is restricted to ideals containing \(k\) as their highest numbered element. When no ambiguity arises, we omit from the matrix notation \(b\) and \(w(S)\). The \((p, w)\)-th element of matrix \(M(S)\) \((M'(S, k))\) is then abbreviated as \(m_{p,w}(S)\) \((m'_{p,w}(S, k))\).

We assume, without loss of generality, that the elements of \(N\) have been numbered so that \(L_1 = 1, 2, \ldots, n\) is one of two defining linear extensions of \(P\), while the other one is denoted by \(L_2\). Accordingly \(i \prec_P k\) iff \(i \prec_{L_2} k\) and \(i \prec_{L_2} k\). We will use the notation \(i||k\) if \(i < k\) as numbers and \(i\) is not comparable to \(k\) in \(P\). The principal ideals are defined by \(B_k = \{i \mid i \preceq_P k\}\) and their ‘complements’ by \(\overline{B}_k = \{i \mid i \mid|k\}\) for \(k = 1, 2, \ldots, n\). Partition the ideals of \(P\) by their highest numbered element, and let \(I_k\) be the set of ideals with highest numbered element \(k\). If \(I \in I_k\), then we clearly must have \(B_k \subseteq I\) and \(I \setminus B_k\) must be an ideal in the induced subposet \(\overline{B}_k\). Therefore, each ideal \(I \in I_k\) is in a one-to-one correspondence with the ideal \(I \setminus B_k\) of the subposet \(\overline{B}_k\) and \(p(I) = p(I \setminus B_k) + p(B_k)\) and \(w(I) = w(I \setminus B_k) + w(B_k)\). Thus, we have proved the following lemma.

**Lemma 1.** Consider a POK problem on the poset \(P\) with knapsack capacity \(b\) and total item value \(W\). If \(N_k = \{1, 2, \ldots, k\}\), and \(M'(N_k, k, b, w(N_k))\) \(M(\overline{B}_k, b, w(N_k) - w(B_k))\) are the two feasibility matrices of interest, then for their corresponding elements we have \(m'_{p,w}(N_k, k) = m_{p - p(B_k), w - w(B_k)}(\overline{B}_k)\) for \(1 \leq p \leq b\) and \(1 \leq w \leq w(N_k)\).

Since \((N, b, W)\) contains a 1 in position \((p, w)\) if and only if there is a matrix \(M'(N_k, k, b, w(N_k))\) containing a 1 in the same position, \(M(N, b, W)\) can be obtained in \(O(bWn)\) time if we know the \(M'(N_k, k, b, w(N_k))\) matrices for \(k = 1, 2, \ldots, n\). If the computation in Lemma 1 was recursively applied to the subposets \(\overline{B}_k\), this would yield an exponential in \(n\) computation of \(M(N, b, W)\) for general posets \(P\). As the following theorem shows, however, the recursion does not need to go beyond the second level if \(\text{dim} P = 2\), i.e., it results in a polynomial-time computation.

**Theorem 1.** If \(\text{dim} P = 2\) for a POK problem with knapsack capacity \(b\) and total item value \(W\), then \(M(N, b, W)\) can be computed by a pseudopolynomial algorithm in \(O(n^2bW)\) time.
Proof. Partition the set of ideals in $P$ by their highest numbered element and apply Lemma 1. Accordingly we obtain a general element of $M(N, b, W)$ by

$$m_{p,w}(N) = \max_{k=1,2,\ldots,n} m'_{p,w}(N_k,k) = \max_{k=1,2,\ldots,n} m_{p-p(B_k),w-w(B_k)}(\overline{B}_k)$$

for $1 \leq p \leq b$ and $1 \leq w \leq W$. We will show that a single computation of $M(\overline{B}_k,b,w(N_k) - w(B_k))$ for a fixed $k$ can be carried out in $O(nbW)$ time. Then the entire computation for $M(N,b,W)$ needs no more than $O(n^2bW)$ time.

Let $C_{kj} = B_k \cap B_j = \{i \mid i \in B_k, i \leq_p j\}$ and $\overline{C}_{kj} = B_k \cap \overline{B}_j = \{i \mid i \in B_k, i < j, i \not\leq_p j\}$ for $k = 1, 2, \ldots, n$ and $j\|k$.

We claim that $dim P = 2$ and $j\|k$ imply $\overline{C}_{kj} = \overline{B}_j$: If $i \in \overline{C}_{kj}$, then it can easily be seen that $i \in \overline{B}_j$. For the other direction, if $i \in \overline{B}_j$, then $i < j$ and $i \not\leq_p j$, i.e., $j \prec_L i$. Furthermore, $j\|k$ implies $j < k$ and $j \not\leq_p k$, i.e., $k \prec_L j$ too, so that by transitivity of $L_k$ $k \prec_L i$ also holds, which implies $i \in \overline{B}_k$, and thus $i \in \overline{C}_{kj}$.

Let us compute $M(\overline{B}_k,b,w(N_k) - w(B_k))$, i.e., the feasibility matrix for the ideals of $\overline{B}_k$, in ascending order of $k = 1, 2, \ldots, n$. For this it suffices to calculate the matrices $M'(\overline{B}_k,j)$ for all $j : j\|k$, since $m_{p,w}(\overline{B}_k) = \max_{j\|k} \{m'_{p,w}(\overline{B}_k,j)\}$. Applying Lemma 1 to the poset $\overline{B}_k$ shows that $M'(\overline{B}_k,j)$ can be derived from $M(\overline{C}_{kj})$ by $m'_{p,w}(\overline{B}_k,j) = m_{p-p(C_{kj}),w-w(C_{kj})}(\overline{C}_{kj})$. But $\overline{C}_{kj} = \overline{B}_j$, therefore we need only the previously computed matrices $M(\overline{B}_j)$. Combining these observations, we obtain

$$m_{p,w}(\overline{B}_k) = \max_{j\|k} \{m'_{p,w}(\overline{B}_k,j)\} = \max_{j\|k} \{m_{p-p(C_{kj}),w-w(C_{kj})}(\overline{C}_{kj})\}$$

which equals $\max_{j\|k} \{m_{p-p(C_{kj}),w-w(C_{kj})}(\overline{B}_j)\}$. Since the quantities $p(B_k)$, $w(B_k)$, $p(C_{kj})$ and $w(C_{kj})$ can all be calculated in $O(n^3)$ time in a preprocessing step, the claimed complexity of the algorithm follows. \hfill $\Box$

The algorithm given in Theorem 1 is polynomial in $W$ and $b$. We show how to compress the table size and find a near-optimal solution instead of the exact optimum. We will obtain a bicriteria FPTAS, i.e., an algorithm that yields an ideal of weight and processing time within $(1 + \varepsilon)$ of the desired targets, in time polynomial in $1/\varepsilon$.

We will scale the coefficients in a manner similar to the one used for standard Knapsack (cf. [11]). The difference is that in our case both $b$ and $W$ appear in the running time so we have to scale both weights and processing times. Let $w_{\text{max}} = \max_k w_k$. Let $k_w = \varepsilon w_{\text{max}}/n$, $p_k = \varepsilon w_k/n$. Set $w' = \lfloor w_j/k_w \rfloor$, $p' = \lfloor p_j/k_p \rfloor \\forall j \in N$. In the scaled instance set the knapsack capacity to be $b' = \lfloor p(N)/(k_pb) \rfloor$. It can be shown that solving the scaled instance yields the following theorem.

**Theorem 2.** For a POK problem with knapsack capacity $b$ on poset $P = (N, \prec_P)$ where $dim P = 2$ there is an FPTAS with the following properties. For any $\varepsilon > 0$ it produces, in time $O(n^5/\varepsilon)$, an ideal of processing time at most $(1 + \varepsilon)b$ and weight at least $(1 - \varepsilon)$ times the optimum.
3 POK on Bipartite Partial Orders

Let \((X, Y; \prec)\) be a bipartite poset with comparability graph \(G = (X, Y; E)\). Its bipartite complement \(\overline{G} = (X, Y; \overline{E})\) is defined by \(\overline{E} = X \times Y \setminus E\). An ideal \(X' \cup Y'\), where \(X' \subseteq X\) and \(Y' \subseteq Y\), is \(Y\)-maximal if there is no \(y \in Y\) such that \(X' \cup Y' \cup \{y\}\) is also an ideal. The ideal \((X', Y')\) is \(X\)-minimal if there is no \(x \in X'\) such that \((X' \setminus x, Y')\) is also an ideal.

Each ideal \(I\) of a poset is uniquely defined by its maximal elements \(\max I = \{a \in I \mid \exists b \in I \text{ such that } a \prec b\}\). If we know \(\max I\), then \(I = \{a \mid \exists b \in \max I \text{ such that } a \preceq b\}\). The elements of \(\max I\) always form an antichain in the poset, and an antichain generates an ideal in this sense. Let \(\mathcal{M} = \{I \mid I\text{ is an ideal such that } \max I\text{ is a maximal antichain}\}\). It is well known that \((\mathcal{M}, \subseteq)\) is a lattice which is isomorphic to the lattice of maximal antichains.

**Lemma 2.** The following statements are equivalent in a bipartite poset \((X, Y; \prec)\) with comparability graph \(G = (X, Y; E)\):

1. \(X' \cup Y'\) is a \(Y\)-maximal and \(X\)-minimal ideal
2. \(Y' \cup (X \setminus X')\) is a maximal antichain
3. The induced subgraph \(\overline{G}[X \setminus X', Y']\) is a maximal complete bipartite subgraph of the bipartite complement \(\overline{G}\).

**Proof.**

1. \(\implies\) 2. Suppose \(X' \cup Y'\) is a \(Y\)-maximal and \(X\)-minimal ideal. We cannot have any edge between \(X' \setminus X'\) and \(Y'\) in \(G\), since \((X', Y')\) is an ideal. Thus \(X' \cup Y' \setminus X'\) is an antichain. Since \(Y'\) is \(Y\)-maximal, every \(y \in Y \setminus Y'\) must have a predecessor in \(X' \setminus X'\), so there is no \(y \in Y \setminus Y'\) such that \(Y' \cup (X \setminus X') \cup y\) would also be an antichain. Similarly, every \(x \in X'\) must have a successor \(y \in Y'\), since \((X', Y')\) is \(X\)-minimal, so there is no \(x \in X'\) for which \(Y' \cup (X \setminus X') \cup x\) would also be an antichain. This proves the maximality of the antichain \(Y' \cup (X \setminus X')\).  

2. \(\implies\) 3. obvious.

3. \(\implies\) 1. Let \(\overline{G}[X \setminus X', Y']\) be a maximal complete bipartite subgraph of \(\overline{G}\). This implies that no \(x \in X \setminus X'\) can be a predecessor for any \(y \in Y'\), thus \((X', Y')\) is an ideal in \((X, Y; \prec)\). Since \(\overline{G}[X \setminus X', Y']\) is a maximal complete bipartite subgraph of \(\overline{G}\), no \(y \in Y \setminus Y'\) is connected to every element of \(X \setminus X'\) in \(\overline{G}\), so there is no \(y \in Y \setminus Y'\) extending the ideal \(X' \cup Y'\), i.e., \(X' \cup Y'\) is \(Y\)-maximal. Similarly, no \(x \in X'\) is connected to every element of \(Y'\) in \(\overline{G}\), so \(X' \cup Y'\) is also \(X\)-minimal.

It is clear that in order to solve a Red-Blue bipartite instance of POK, we need to search through only the ideals which are \(Y\)-maximal and \(X\)-minimal, i.e., the ideals in \(\mathcal{M}\). By the above lemma, these ideals of a bipartite poset \((X, Y; \prec)\) are in a one-to-one correspondence with the maximal complete bipartite subgraphs of the bipartite complement of its comparability graph. If we have such a subgraph \(\overline{G}[U, Z]\), then the corresponding ideal \(X \setminus U \cup Z\), its weight \(w(X \setminus U \cup Z)\) and processing time \(p(X \setminus U \cup Z)\) can all be computed in \(O(n)\) time. Furthermore, if \(\overline{G}\) is chordal bipartite, then it has only at most \(|\overline{E}|\) maximal complete bipartite subgraphs and Kloks and Kratsch [14] found an algorithm which lists these in \(O(|X \cup Y| + |\overline{E}|)\) time if \(\overline{G}\) is given by an appropriately ordered version of its bipartite adjacency matrix. This proves the following theorem.
Theorem 3. Consider a Red-Blue instance of POK on a bipartite poset \((X, Y; \prec)\) with comparability graph \(G\). If \(\overline{G}\) is chordal bipartite then there is an algorithm which solves this POK problem in \(O(n^3)\) time.

We proceed now to lift the restriction that the bipartite instance is Red-Blue. The resulting problem is NP-hard since it contains classical Knapsack. Let \(I = B \cup C\) be an arbitrary ideal in the bipartite poset \((X, Y; \prec)\), where \(B \subseteq X\) and \(C \subseteq Y\). It is clear that \(\max I\) partitions into \(C \cup (\max I \cap B)\).

Let \(X' = B \setminus \max I\) and \(Y' = \{ y \in Y \mid \exists x \in X \setminus X'\text{ such that }x \prec y\}\). It is clear that \(C \subseteq Y'\) and that \(X' \cup Y'\) is \(Y\)-maximal and \(X\)-minimal. Then by Lemma 2, \(Y' \cup (X \setminus X')\) is a maximal antichain containing \(\max I\). The ideal generated by this maximal antichain is \(Y' \cup X \in \mathcal{M}\). Furthermore, \(I \subseteq (Y' \cup X)\) and \((Y' \cup X) \setminus I\) is an antichain contained in \(Y' \cup (X \setminus X')\). Thus \(I\) can be derived from \((X \cup Y')\) by the deletion of an appropriate unordered subset of \(Y' \cup (X \setminus X')\).

If we considered for deletion all such subsets of \(Y' \cup (X \setminus X')\), and repeated this for all \((X \cup Y') \in \mathcal{M}\), then we would derive every ideal of the poset \((X, Y; \prec)\) (some of them possibly several times.)

Consider now an arbitrary instance of POK on a bipartite poset \((X, Y; \prec)\) with knapsack size \(b\) and optimal solution weight \(W^*\). If there exists an \((X \cup Y') \in \mathcal{M}\) such that \(p(X \cup Y') \leq b\), then clearly \(W^* = \max \{w(X \cup Y') \mid p(X \cup Y') \leq b, (X \cup Y') \in \mathcal{M}\}\). Otherwise, consider an infeasible \((X \cup Y') \in \mathcal{M}\), i.e., \(p(X \cup Y') > b\). Define \(X' = \{ x \in X \mid \exists y \in Y'\text{ such that }x \prec y\}\). We can find the largest-weight feasible ideal derivable from \((X \cup Y')\) by the above process by solving the auxiliary Knapsack problem \(\{\max w(J) \mid J \subseteq ((X \setminus X') \cup Y'), p(J) \leq b - p(X')\}\).

The optimal solution \(J^*\) of this can be found by a pseudopolynomial algorithm in \(O(nW)\) time, and \(X' \cup J^*\) is the largest-weight feasible ideal that can be derived from this \((X \cup Y')\) for the original POK instance. As we have discussed earlier, if \((X, Y; \prec)\) is a bipartite poset whose comparability graph has a bipartite complement \(\overline{G} = (X, Y; \overline{E})\) which is chordal bipartite, then \(|\mathcal{M}| \leq |\overline{E}|\), so that we need to call the pseudopolynomial algorithm for the solution of at most \(|\overline{E}|\) auxiliary problems. This proves the following.

Theorem 4. Consider an instance of POK on a bipartite poset \((X, Y; \prec)\) with comparability graph \(G\). If \(\overline{G}\) is chordal bipartite then there is a pseudopolynomial algorithm which solves this POK problem in \(O(n^3W)\) time.

It is easy to see that invoking as a Knapsack oracle the FPTAS in [12], instead of a pseudopolynomial algorithm, yields a \((1 - \varepsilon)\) weight-approximation to the original problem.

Theorem 5. Consider an instance of POK on a bipartite poset \((X, Y; \prec)\) with comparability graph \(G\). If \(\overline{G}\) is chordal bipartite then there is an FPTAS which for any \(\varepsilon > 0\), solves this POK problem in \(O(n^3 \log(1/\varepsilon) + n^2(1/\varepsilon^4))\) time.

4 Applications to Scheduling

In this section we show how the above pseudopolynomial algorithms for POK given in Theorems 1 and 4 lead to improved polynomial-time approximation
algorithms for special cases of the scheduling problem $1|\text{prec}|\sum w_j C_j$. In a recent paper, Woeginger [21] defined the following auxiliary problem, which is a special case of POK.

**Problem: GOOD INITIAL SET (IDEAL)**

**Instance:** An instance of $1|\text{prec}|\sum w_j C_j$ with nonnegative integer processing times $p_i$ and weights $w_i$, a real number $\gamma$ with $0 < \gamma \leq 1/2$.

**Question:** Is there an ideal $I$ in the poset $P$ representing the precedence constraints for which $p(I) \leq (1/2 + \gamma)p(N)$ and $w(I) \geq (1/2 - \gamma)w(N)$?

The following theorem shows the strong connection between the solvability of POK and the approximability of $1|\text{prec}|\sum w_j C_j$. Its derivation uses 2-dimensional Gantt charts as introduced in [9].

**Theorem 6.** [21] If $\mathcal{C}$ is a class of partial orders on which GOOD INITIAL SET is solvable in pseudopolynomial time, then for any $\epsilon > 0$, the restriction of the scheduling problem to $\mathcal{C}$, i.e., $1|\text{prec}, \mathcal{C}|\sum w_j C_j$ has a polynomial-time $(\Phi + \epsilon)$-approximation algorithm, where $\Phi = 1/2(\sqrt{5} + 1) \approx 1.61803$.

Theorems 1, 4 and 6 yield the following corollaries (proofs omitted).

**Corollary 1.** For any $\epsilon > 0$, $1|\text{prec}, \text{dim}P = 2|\sum w_j C_j$ has a polynomial-time $(\Phi + \epsilon)$-approximation algorithm.

**Corollary 2.** For any $\epsilon > 0$, the problem $1|\text{prec}, \prec_P|\sum w_j C_j$ where $P = (X,Y;\prec_P)$ is such that the bipartite complement of its comparability graph is a chordal bipartite graph, has a polynomial-time $(\Phi + \epsilon)$-approximation algorithm.

## 5 The LP-Relaxation for General POK

In this section we examine the natural integer program for POK and the associated linear relaxation. We give an algorithm for rounding a fractional solution which satisfies weight-majority. Consider the following integer program with parameter $l > 1$:

$$\max \left\{ \sum_{j \in N} w_j x_j \mid \sum_{i \in N} p_i x_i \leq p(N)/l, \ x_j \leq x_i \ i < j, \ x_i \in \{0,1\} \ i \in N \right\}.$$  

Relaxing the integrality constraint gives a linear relaxation which we denote by $LP(l)$. Let $\bar{x}$ be a solution to $LP(l)$ with objective value $\sum_{j \in N} w_j \bar{x}_j = w(N)/h$, where $h < l$. We show how to round it to an integral solution, while relaxing by a small factor the right hand side of the packing constraint. The guaranteed value of the objective will also be less than $w(N)/h$, therefore producing a bicriteria approximation. The algorithm itself is simple. Let $\alpha \in (0,1)$ be a parameter of our choice. We apply filtering based on $\alpha$. We omit the analysis in this version.

**Theorem 7.** Let a POK instance be such that the linear relaxation $LP(l)$ has a solution $\bar{x}$ of value $w(N)/h$ with $1/l < 1/h$. Define $N' = \{ j \in N \mid \bar{x}_j \geq \alpha \}$. $N'$ is an ideal and for any $\alpha \in (1/l,1/h)$ $w(N') \geq \frac{1-h\alpha}{\alpha(1-\alpha)}w(N)$ and $p(N') \leq \frac{w(N)}{\alpha l}$.  

Partially-Ordered Knapsack and Applications to Scheduling

Theorem 7 raises two questions. First, when can we expect the optimal fractional solution to meet the weight-majority condition $1/l < 1/h$? Second, the guarantee on the weight does not conform to the standard definition of a multiplicative $\rho$-approximation. Can we obtain a proper $\rho$-approximation on the weight with or without relaxing the upper bound on the processing time? We discuss these two questions in the next two sections.

6 The Effect of Decomposability

In this section we address the question: when can one hope to obtain a weight majority solution to $LP(l)$ and hence apply Theorem 7? We show that decomposability of the input is a necessary condition by gaining insight into the structure of LP solutions. Finally we show, non-constructively, an upper bound on the integrality gap of $LP(l)$.

To simplify our derivation we assume the input instance is bipartite Red-Blue. This is without loss of generality: given a general input $N_0$ one can transform it to a bipartite input by having for each job $i_0 \in N_0$ a vertex $i, i'$ on each side $X, Y$ of the partition. The vertex $i$ on the $X$-side assumes the processing time and $i'$ the weight of $i_0$. Precedence constraint $i_0 \prec j_0$ for $i_0, j_0 \in N_0$ translates to $i \prec j'$. Moreover $i \prec i'$ for all $i$. Without loss of generality we can assume that in any ideal of $N$, inclusion of $i$ implies inclusion of $i'$. Thus there is a one-to-one correspondence between ideals of $N$ and $N_0$, with the total processing time and weight being the same.

For any $l > 1$ formulation $I Pok$, defined below, is the integer linear programming formulation of the associated partially-ordered knapsack instance.

$$\max \sum_{j \in Y} w_j x_j \sum_{i \in X} p_i x_i \leq p(X)/l, \quad x_j - x_i \leq 0 \ i \prec j, \quad x_i \in \{0, 1\} \ i \in X \cup Y.$$

In the linear programming relaxation, $LPok$, the integrality constraints are relaxed. The proofs of the following two theorems are omitted.

Theorem 8. The solution $x_i = 1/l$ for $i \in X \cup Y$ is optimal for $LPok$ with knapsack capacity $p(X)/l$ if and only if the underlying poset is indecomposable.

Theorem 8 shows that when the input is indecomposable, the same fraction of weight and processing time will be assigned to the ideal by the optimal solution to $LPok$, hence the algorithm from Theorem 7 cannot apply. However, Theorem 8 does not express only the limitations of our algorithm but the limitations of the relaxation as well. The linear program $LPok$ can be seen as maximizing the rank of the ideal that meets the processing-time packing constraint. When the balanced solution in which all variables are equal is optimal, in the case of indecomposability, the rank computed is equal to $w(Y)/p(X)$, which is equivalent to selecting the entire set $X \cup Y$ as the ideal. Therefore no information is really obtained from $LPok$ in this case.

Theorem 9. The fractional optimum of $LPok$ is at most $(\lambda/l) w(Y)$ on an input with maximum-rank ideal of value $\lambda w(Y)/p(X), \lambda \geq 1$.
Finally we observe that the valid inequalities for Knapsack (without precedence constraints) recently proposed by Carr et al. \cite{2} do not seem to help in the difficult case, namely indecomposability. If we augment LPOK with these inequalities, it is easy to see that the balanced solution is feasible for the new formulation as well.

7 Evidence of Hardness

The positive result of Theorem 7 has two limitations: first it applies only to fractional solutions satisfying weight-majority and second it does not give a $\rho$-approximation for the weight in the classical multiplicative sense. In this section we show that overcoming these limitations would lead to improved approximations for both the time-majority case and $1|\text{prec}| \sum w_j C_j$. This would make progress on a longstanding open problem. For the purposes of this section a $\rho$-approximation algorithm, $\rho < 1$, for a POK instance $(N, h, l)$ yields, if the instance is feasible, an ideal of weight at least $\rho w(N)/h$ and processing time at most $p(N)/l$. The idea behind the upcoming theorem is that even if a given set $N$ has no ideal $N'$ such that $w(N')/w(N) > p(N')/p(N)$, we can add auxiliary weight so that a feasible weight-majority instance $I'$ can be defined. If in turn we have an algorithm to extract from $I'$ an ideal with large enough weight, this answer will contain a good fraction of the original weight from $N$. The proof of the theorem is omitted.

**Theorem 10.** Let $\delta$ be any constant in $(0, 1)$. If there is a $\delta$-approximation algorithm $A$ for weight-majority knapsack instances, then there exists an $O(1)$-approximation algorithm for a time-majority instance $(N, h, l)$ where $h$ and $l$ are constants depending on $\delta$.

Theorem 10 shows that finding an $O(1)$-approximation for the weight-majority case is as hard as finding an $O(1)$-approximation for a class of time-majority instances with small but constant $1/h$ and $1/l$ parameters. This in turn is a highly non-trivial problem. Its solution would lead to a $2 - \beta$ approximation for some fixed $\beta$ for $1|\text{prec}| \sum w_j C_j$, however small $1/h$ and $1/l$. The latter statement can easily be shown using 2-D Gantt charts \cite{9} and methods similar to the ones used in \cite{21}. We omit the details.

**Corollary 3.** Let $\delta$ be any constant in $(0, 1)$. If there is a $\delta$-approximation algorithm for weight-majority knapsack instances, there is a $2 - \beta$ approximation for some fixed $\beta$ for $1|\text{prec}| \sum w_j C_j$.

Acknowledgment

Stavros Kolliopoulos thanks Gerhard Woeginger for useful discussions.
References


