Precedence-Constrained Min Sum Set Cover*

Jessica McClintock[1], Julián Mestre[2], and Anthony Wirth[†3]

1 **School of Computing and Information Systems, The University of Melbourne, Parkville, Australia**
jessica.mcclintock@unimelb.edu.au

2 **School of Information Technologies, The University of Sydney, Darlington, Australia**
julian.mestre@sydney.edu.au

3 **School of Computing and Information Systems, The University of Melbourne, Parkville, Australia**
awirth@unimelb.edu.au

Abstract

We introduce a version of the Min Sum Set Cover (MSSC) problem in which there are "AND" precedence constraints on the m sets. In the Precedence-Constrained Min Sum Set Cover (PCMSSC) problem, when interpreted as directed edges, the constraints induce an acyclic directed graph. PCMSSC models the aim of scheduling software tests to prioritize the rate of fault detection subject to dependencies between tests.

Our greedy scheme for PCMSSC is similar to the approaches of Feige, Lovász, and, Tetali for MSSC, and Chekuri and Motwani for precedence-constrained scheduling to minimize weighted completion time. With a factor-4 increase in approximation ratio, we reduce PCMSSC to the problem of finding a maximum-density precedence-closed sub-family of sets, where density is the ratio of sub-family union size to cardinality. We provide a greedy factor-$\sqrt{m}$ algorithm for maximizing density; on forests of in-trees, we show this algorithm finds an optimal solution. Harnessing an alternative greedy argument of Chekuri and Kumar for Maximum Coverage with Group Budget Constraints, on forests of out-trees, we design an algorithm with approximation ratio equal to maximum tree height.

Finally, with a reduction from the Planted Dense Subgraph detection problem, we show that its conjectured hardness implies there is no polynomial-time algorithm for PCMSSC with approximation factor in $O(m^{1/12-\varepsilon})$.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems

Keywords and phrases planted dense subgraph, min sum set cover, precedence constrained

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.55

1 Introduction

In this paper, we introduce the PRECEDENCE-CONSTRAINED MIN SUM SET COVER problem, which has connections to MIN SUM SET COVER, DENSEST SUBGRAPH, PRECEDENCE-CONSTRAINED SCHEDULING TO MINIMIZE TOTAL WEIGHTED COMPLETION TIME, SCHEDULING WITH AND/OR PRECEDENCE CONSTRAINTS, and several other problems.

* This work was partially supported by the Australian Research Council.
† Corresponding author. Anthony Wirth was supported by an Australian Research Council Future Fellowship.

Problem definition. Just like SET COVER, or indeed MIN SUM SET COVER (aka MSSC), the input to PRECEDENCE-CONSTRAINED MIN SUM SET COVER (aka PCMSSC) is a family of sets, $\mathcal{F}$, whose union is the *universe*, U. In addition, there is a binary precedence relation on $\mathcal{F}$, represented by $\prec$. Let $G(\mathcal{F}, \prec)$ stand for the directed graph that $\prec$ induces on $\mathcal{F}$. Throughout this presentation, we assume G is *acyclic.* The output is a permutation, π, of the family of sets, $\mathcal{F}$, that obeys the precedence relation $\prec$. That is, if $A \prec B$, then A must precede B in the permutation: $\pi^{-1}(A) < \pi^{-1}(B)$. The objective value, to be minimized, is the sum over every item in U of its *first* covering time in the permutation. That is, for each item u, let $\tau(u) = \min_{A\in\mathcal{F}}\{\pi^{-1}(A) : u \in A\}$ be the index of the earliest set in π that includes item u; the objective is $\sum_{u\in U} \tau(u)$. For convenience, let n be the number of items in the universe $|U|$, and let m be the cardinality of the family of sets $|\mathcal{F}|$, and index the permutation π from 1 to m.

1.1 Application

In the software testing context, we would like to schedule test sequences to prioritize the rate of fault detection. However, there may be inherent dependencies between the tests – some test cases need to be scheduled before others – complicating the process of ordering the test suite [18]. Though they perform well, existing algorithms for test case prioritization subject to dependencies, are heuristic in their effectiveness [18, 17]. The PCMSSC problem crystallizes the aims and constraints of this software test prioritization problem in a way that admits analysis and approximation, yet is realistic. PCMSSC is a small extension of one existing combinatorial optimization question, MSSC, and a refinement of another, SCHEDULING WITH AND/OR PRECEDENCE CONSTRAINTS (aka SAOPC).

1.2 Theoretical context

The original MSSC problem has the same objective as PCMSSC, to minimize $\sum_{u\in U} \tau(u)$, but it permits every ordering π of $\mathcal{F}$. Feige, Lovász and Tetali's greedy algorithm for MSSC [10] starts from an empty ordering π and is simply: *While* $|\pi| < m$*, append to* π *a set maximizing the number of (yet) uncovered items.* Via a clever pricing and histogram argument, they show that this is a 4-approximation; they also show that this is the best possible unless P equals NP.

Chekuri and Motwani attack the PRECEDENCE-CONSTRAINED SCHEDULING TO MINIMIZE TOTAL WEIGHTED COMPLETION TIME (aka PCSTW) problem, which has the same precedence structure as PCMSSC. Here, however, the total weighted completion times objective is additive, whereas set coverage is (only) monotone submodular. Chekuri and Motwani's factor-2 algorithm for PCSTW repeatedly (and optimally) solves the subproblem MINIMUM-RANK PRECEDENCE-CLOSED SUBGRAPH (aka MRPCS) [7]. The *rank* of a family of jobs is the ratio of its total processing time to its total weight.

Key sub-problem definition. To produce an approximation algorithm for PCMSSC, we combine the ideas of Feige et al. and Chekuri and Motwani. We study a problem we call MAX-DENSITY PRECEDENCE-CLOSED SUBFAMILY (aka MDPCS): in some sense, *density* is the reciprocal of rank. We let the *coverage* of sub-family, $\mathcal{A}$ of $\mathcal{F}$, be the union of the sets in the sub-family: $\text{cov}(\mathcal{A}) \equiv \cup_{A\in\mathcal{A}} A$. Often, we consider the coverage of a sub-family on some subset X of the universe: $\text{cov}(\mathcal{A}, X) = \text{cov}(\mathcal{A}) \cap X$. The density, Δ, of a non-empty sub-family on subset X is the ratio of the size of its coverage to its cardinality: $\Delta(\mathcal{A}, X) \equiv |\text{cov}(\mathcal{A}, X)|/|\mathcal{A}|$. (When it is obvious, we omit the second argument of Δ.)

Algorithm 1 Algorithm PCMSSC-Greedy.

```
1: function PCMSSC-Greedy($\mathcal{F}, \prec$)
2:     $\pi \leftarrow$ an “empty permutation”
3:     $\mathcal{G} \leftarrow \mathcal{F}$; $R \leftarrow U$
4:     while $R \neq \varnothing$ do
5:         $\mathcal{A} \leftarrow D(\mathcal{G}, \prec, R)$
6:         Append to $\pi$ some permutation of $\mathcal{A}$ consistent with $\prec$
7:         $\mathcal{G} \leftarrow \mathcal{G} \setminus \mathcal{A}$; $R \leftarrow R \setminus \mathrm{cov}(\mathcal{A})$
8:     Return $\pi$
```

For convenience' sake, $\Delta(\varnothing, X)$ is defined to be negative. The MDPCS problem seeks a sub-family $\mathcal{A}$ of some input family of sets $\mathcal{G}$ that maximizes density on a remaining set of items to be covered, R, with $R \subseteq \mathrm{cov}(\mathcal{G})$, and is *precedence closed*. That is, the aim is to maximize $\Delta(\mathcal{A}, R)$, with the requirement that if $A \prec B$ and $B \in \mathcal{A}$, then $A \in \mathcal{A}$. Were sets in $\mathcal{G}$ pairwise disjoint, we could adopt Chekuri and Motwani's approach for MRPCS, but in general, it is highly unlikely that such a polynomial-time optimal algorithm exists for MDPCS. Indeed, our hardness-of-approximation result for MDPCS arises from a connection to Densest k-Subgraph, whereas Chekuri and Motwani's max-flow algorithm solves MRPCS in polynomial time, similar to the approach for Densest Subgraph [12].

1.3 Our results

We first show that an approximately good solution to MDPCS provides, within factor 4, an approximately good solution to PCMSSC.

We describe a greedy algorithm, MDPCS-Greedy, for MDPCS that obtains a $\sqrt{m}$ approximation, and show that (up to a factor 2) this analysis is tight. We extend MDPCS-Greedy to be iteratively greedy, and again show that $O(\sqrt{m})$ is the best approximation we can obtain. If the precedence relation, $\prec$, induces a forest of in-trees, we show that MDPCS-Greedy in fact solves MDPCS optimally in polynomial time. If the precedence relation, $\prec$, induces a forest of out-trees, we introduce a polynomial-time approximation with factor equal to the largest tree *height*.

Consistent with the large approximation factors found in our algorithms, we show there is no approximation algorithm for PCMSSC with factor in $O(m^{1/12-\varepsilon})$. This result assumes the Planted Dense Subgraph Conjecture (aka PDSC), which states that it is hard to find inside an Erdős-Rényi graph a planted dense E-R component. Recently, hardness of approximation of the Target Set Selection problem was shown via a reduction from Planted Densest Subgraph and its conjectured hardness [5].

2 Reduction to Max-Density Precedence-Closed Subfamily

In this section, we show that PCMSSC *reduces* to MDPCS. Suppose we have an algorithm, D, that returns a factor-α approximation solution to MDPCS. Consider the greedy scheme in Algorithm 1 for PCMSSC, which we call PCMSSC-Greedy.

Since D is a polynomial-time algorithm (and X only gets smaller), since the while loop runs at most n times, and since topological sorting takes polynomial time, PCMSSC-Greedy runs in polynomial time. We now prove that this scheme is in fact an approximation algorithm.

Although factor α might be a function of both m and n, we assume α is monotonically non-decreasing in both, so we can safely let α stand for $\alpha(m, n)$ in the following.

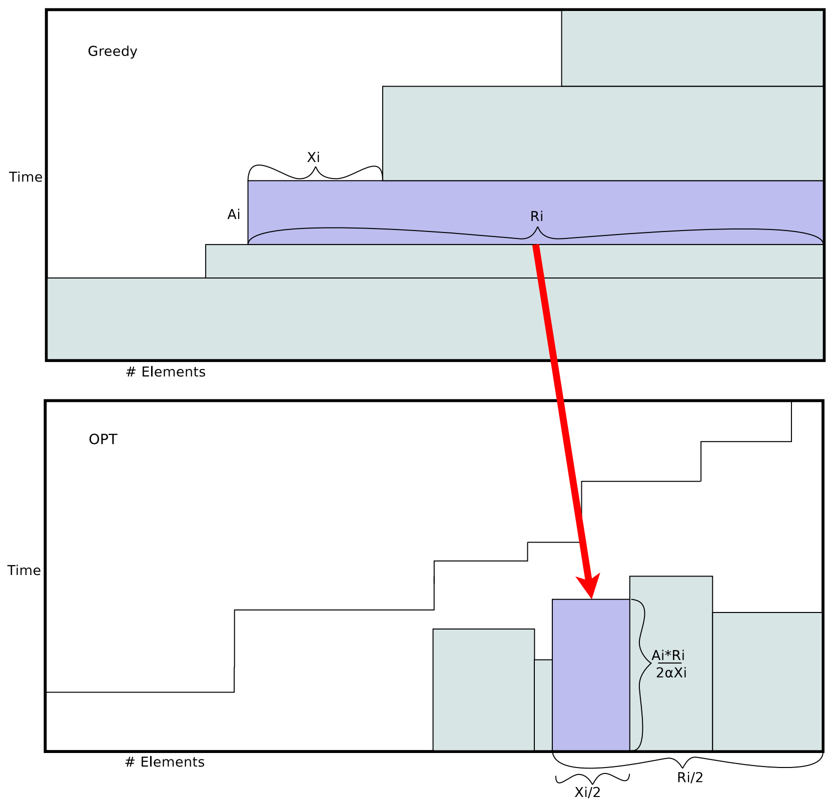

Figure 1 Mapping from upper-bound plot $\mathcal{B}$ to one with area a factor $4 \cdot \alpha$ smaller.

▶ **Lemma 1.** PCMSSC-Greedy *is a* $4 \cdot \alpha$ *approximation algorithm to* PCMSSC.

Proof. Let $\mathcal{A}_i$ be the sub-family returned by D in iteration i of PCMSSC-Greedy, and let $m_i = \sum_{j=1}^{i} |\mathcal{A}_j|$. Also, let R_i be the subset of U not yet covered after $i-1$ iterations, $\{u : \tau(u) > m_{i-1}\}$, and let $X_i \subseteq R_i$ be the subset of U that is first covered by some set in $\mathcal{A}_i$, $\{u : m_{i-1} < \tau(u) \leq m_i\}$.

We can upper-bound the cost of PCMSSC-Greedy by $\sum_i |R_i||\mathcal{A}_i|$. At worst, each of the X_i items is covered by the last set in $\mathcal{A}_i$, number m_i; hence, at iteration i, appending (a permutation of) $\mathcal{A}_i$ to π increases the cover time of all items in R_i by (at most) $|\mathcal{A}_i|$.

To prove the approximation factor, we adapt the argument of Feige et al. [10]. Consider a plot of cover time against item number, where we order $u \in U$ by $\tau(u)$ (Figure 1).

That is, on $(u-1, u]$ the plot has height $\tau(u)$, and the plot is non-decreasing on $(0, n]$. The PCMSSC solution cost is the area under plot on $(0, n]$. The upper bound for the PCMSSC-Greedy solution in the previous paragraph can be viewed as a series of horizontal slices, of height $|\mathcal{A}_i|$ and width $|R_i|$, with slices "right-aligned". That is, slice i is the rectangle $(n - |R_i|, n] \times (m_{i-1}, m_i]$. The plot defined by the upper boundary of this series of slices, which we call $\mathcal{B}$, lies not below the plot for PCMSSC-Greedy. We show that, with area shrunk by factor of $1/(4\alpha)$, a mapping of plot $\mathcal{B}$ lies not above the plot for (every) optimal solution, OPT, on $(0, n]$. Since the area under the plot represents solution cost, we conclude that PCMSSC-Greedy is a $4 \cdot \alpha$ approximation.

Mapping. We map slice i to a column of height $h_i \equiv |\mathcal{A}_i||R_i|/(2\alpha|X_i|)$ and width $|X_i|/2$, positioned between $|R_i|/2$ and $|R_{i+1}|/2$ "elements" from the right-hand end of the curve for

Algorithm 2 Algorithm MDPCS-Greedy.

```
1: function MDPCS-Greedy($\mathcal{G}$, $\prec$, $R$)
2:     $\mathcal{A} \leftarrow \mathcal{G}$
3:     for each $S \in \mathcal{G}$ do
4:         if $\Delta(\mathcal{P}[S], R) > \Delta(\mathcal{A}, R)$ then
5:             $\mathcal{A} \leftarrow \mathcal{P}[S]$
6:     return $\mathcal{A}$
```

OPT. This column is the rectangle $(n - |R_i|/2, n - |R_{i+1}|/2] \times (0, h_i]$. (Sets appended to π after all elements of U are covered do not contribute to the solution cost, so we omit them from this analysis, and thus assume $|X_i| > 0$.) Since $h_i|X_i|/2 = |\mathcal{A}_i||R_i|/(4\alpha)$, this mapping produces a plot whose area is a factor 4α smaller than plot $\mathcal{B}$. The following claim suffices to prove Lemma 1, where OPT$[j]$ is the prefix of j sets in permutation OPT (of $\mathcal{F}$).

▶ **Claim 2.** *For all i with $|R_i| > 0$, $|\text{cov}(\text{OPT}[\lfloor h_i \rfloor], R_i)| \leq |R_i|/2$.*

The proof of Claim 2 follows soon. Meanwhile, finalizing the proof of Lemma 1, Claim 2 shows that even after the first $\lfloor h_i \rfloor$ sets of solution OPT, there are at least $|R_i|/2$ uncovered items (of R_i). Therefore the plot for OPT rises to a height at least $\lfloor h_i \rfloor + 1 \geq h_i$, at a horizontal position at most $n - |R_i|/2$. For all i, the top-left corner of the i^{th} mapped column is at position $(h_i, n - |R_i|/2)$; since the plot for OPT is non-decreasing, this rectangle fits entirely within the plot for OPT, and we have the desired shrunken plot. ◀

Proof of Claim 2. Let $\pi \circ j$ stand for π after $j - 1$ iterations of the loop in PCMSSC-Greedy. Abusing notation (as OPT and π are sequences, not families of sets), if $\text{OPT}[\lfloor h_i \rfloor] \subseteq \pi \circ i$, the claim is trivially true, since $|\text{cov}(\text{OPT}[\lfloor h_i \rfloor], R_i)| = 0$. We hence assume that $\text{OPT}[\lfloor h_i \rfloor] \setminus \pi \circ i$ is non-empty. On its i^{th} instantiation, algorithm D returns a sub-family, $\mathcal{A}_i$, whose density $\Delta(\mathcal{A}_i, R_i)$ is $\geq 1/\alpha$ times the maximum-density precedence-closed subfamily of sets. Since sub-family $\text{OPT}[\lfloor h_i \rfloor]$ is precedence closed with respect to $\mathcal{F}$, so is $\text{OPT}[\lfloor h_i \rfloor] \setminus \pi \circ i$ with respect to $\mathcal{G}_i = \mathcal{F} \setminus \pi \circ i$, and it was thus "considered" by D, so

$$\frac{|\text{cov}(\text{OPT}[\lfloor h_i \rfloor] \setminus \pi \circ i), R_i)|}{|\text{OPT}[\lfloor h_i \rfloor] \setminus \pi \circ i|} \leq \alpha \cdot \frac{|X_i|}{|\mathcal{A}_i|}. \tag{1}$$

Now, R_i is exactly those items not in $\text{cov}(\pi \circ i)$, and $|\text{OPT}[\lfloor h_i \rfloor] \setminus \pi \circ i| \leq \lfloor h_i \rfloor$, so inequality (1) leads to $|\text{cov}(\text{OPT}[\lfloor h_i \rfloor], R_i)| \leq \alpha\, h_i\, |X_i|/|\mathcal{A}_i|$. Substituting $h_i = |\mathcal{A}_i||R_i|/(2\alpha|X_i|)$ into this, we obtain $|\text{cov}(\text{OPT}[\lfloor h_i \rfloor], R_i)| \leq |R_i|/2$. ◀

3 Algorithms for Max-Density Precedence-Closed Subfamily

In this section, we introduce a general greedy method for MDPCS, optimal on in-trees, and an alternative approach for out-tree forests, with factor equal to maximum tree height.

3.1 Greedy

Recall that the input to MDPCS is a family of sets $\mathcal{G}$ and a set of items to be covered R. Let $\mathcal{P}[S]$ be the minimal precedence-closed sub-family of $\mathcal{G}$ containing $S \in \mathcal{G}$: that is, the *ancestors* of S (including S itself). Our greedy algorithm for MDPCS, in Algorithm 2 (which we call MDPCS-Greedy), returns the denser of $\mathcal{G}$ and the best of the $\mathcal{P}[S]$ solutions.

By the definition of $\mathcal{P}$, MDPCS-Greedy returns a feasible solution. We let $\delta_{\mathcal{P}\text{-max}}$ stand for the maximum of $\Delta(\mathcal{P}[S], R)$ over all $S \in \mathcal{G}$. If every set in $\mathcal{G}$ covers at least one item in R, then $\delta_{\mathcal{P}\text{-max}} \geq 1$. However, since PCMSSC-Greedy could involve a sequence of depleting instances of MDPCS, there is no guarantee that each set in $\mathcal{G}$ contains an item in R.

▶ **Lemma 3.** MDPCS-Greedy *is a* $\sqrt{m}$ *approximation to* MDPCS*. If* $\delta_{\mathcal{P}\text{-max}} \geq 1$*,* MDPCS-Greedy *is a* $\sqrt{n}$ *approximation to* MDPCS*.*

Proof. Consider some optimal solution sub-family, OPT, and let k stand for its cardinality. For each $S_1, S_2, \ldots, S_k$ in OPT, the density of $\mathcal{P}[S_i]$ is at most $\delta_{\mathcal{P}\text{-max}}$ and, by definition, $\mathcal{P}[S_i]$ is a sub-family of OPT. Therefore,

$$|\text{cov}(\text{OPT}, R)| = |\cup_{i=1}^{k}\text{cov}(\mathcal{P}[S_i], R)| \leq \sum_{i=1}^{k} |\text{cov}(\mathcal{P}[S_i], R)| \leq \delta_{\mathcal{P}\text{-max}} \sum_{i=1}^{k} |\mathcal{P}[S_i]| \leq \delta_{\mathcal{P}\text{-max}} \cdot k^2 \,,$$

where the first inequality observes the definition of union, while the remarks above justify the second and third inequalities. Therefore $\Delta(\text{OPT}, R)/\delta_{\mathcal{P}\text{-max}} \leq k$.

On the other hand, $\Delta(\text{OPT}, R) \leq |R|/k$, but MDPCS-Greedy returns $\mathcal{A}$ with $\Delta(\mathcal{A}, R) \geq \Delta(\mathcal{G}, R) = |R|/m$. Hence the approximation ratio is at most $\min(k, m/k) \leq \sqrt{m}$. If $\delta_{\mathcal{P}\text{-max}} \geq 1$, then the approximation ratio is at most $\min(k, |R|/k) \leq \sqrt{|R|} \leq \sqrt{n}$. ◀

Indeed, these factors for MDPCS-Greedy are tight, up to a factor two. And without the assumption $\delta_{\mathcal{P}\text{-max}} \geq 1$, there are instance collections on which MDPCS-Greedy achieves only an $\Omega(n)$ approximation. Alternatively, we could *iterate* MDPCS-Greedy, repeatedly choosing a sub-family of the form $\mathcal{P}[S]$ (for some $S \in \mathcal{G}$) that when *added* to the current solution maximizes the density of the sub-family, similar to the greedy algorithm for Set Cover. Again, there is a collection of instances in which this scheme returns only an $O(\sqrt{n})$ (or $O(\sqrt{m})$) approximation.

3.2 Forest of in-trees

If graph $G(\mathcal{F}, \prec)$, and hence graph $G(\mathcal{G}, \prec)$ has a special structure, similarly explored in the context of Partially Ordered Knapsack [15], MDPCS admits better approximation factors. We start with G a forest of in-trees: for all A in $\mathcal{F}$, at most one set immediately depends on A, that is $|\{B \in \mathcal{F} : A \prec B\}| \leq 1$. Consequently, for all $A, B \in \mathcal{F}$, *either* $\mathcal{P}[A]$ and $\mathcal{P}[B]$ are disjoint, *or* (wlog) $A \in \mathcal{P}[B]$. Therefore, a solution in such an input is a union of disjoint sub-families $\mathcal{P}[S_1], \mathcal{P}[S_2], \ldots$. In an optimal solution, each sub-family $\mathcal{P}[S_i]$ has optimum density, so MDPCS-Greedy will (in polynomial time) find an optimal solution.

3.3 Forest of out-trees

We consider the "opposite" scenario, in which the in-degree of each set is at most one: that is, for all A, $|\{B \in \mathcal{F} : B \prec A\}| \leq 1$. Focusing on the graph $G(\mathcal{F}, \prec)$, each connected component of G is a rooted out-tree. Here, we introduce another greedy algorithm, which provides an approximation factor equal to the largest tree height. It acts recursively, adopting the approach of Chekuri and Kumar [6] for the Maximum Coverage Problem with Group Budget Constraints, and so adds 1 to the approximation factor at each tree level.

Let OPT be some optimal solution to MDPCS on $\mathcal{G}$, and let δ_{OPT} be its density, $\Delta(\text{OPT}, R)$: therefore, $\text{cov}(\text{OPT}, R) - \delta_{\text{OPT}}|\text{OPT}| = 0$. Our recursive algorithm $D^{\mathcal{T}}$, shown in Algorithm 3, has as input $(\sigma + 1, T, \delta_{\text{OPT}}, R')$, where $\sigma + 1$ is an "approximation factor", T a tree, and $R' \subseteq R$ a subset of items to be covered. Let $t \sqsubset T$ denote that t is a subtree of T

Algorithm 3 Algorithm $D^{\mathcal{T}}$ called with $(\sigma+1, T, \delta_{\mathrm{OPT}}, R')$.

```
1: function $D^{\mathcal{T}}(\sigma+1, T, \delta_{\mathrm{OPT}}, R')$
2:     $r \leftarrow$ root of $T$
3:     $R'_0 \leftarrow R'$, $\mathcal{A}^0_T \leftarrow \{r\}$
4:     Solutions $\leftarrow \{\varnothing, \mathcal{A}^0_T\}$
5:     $R' \leftarrow R' \setminus \mathrm{cov}(\mathcal{A}^0_T)$
6:     $\kappa \leftarrow$ number of children of $r$
7:     for $j = 1$ to $\kappa$ do
8:         Let $T_j$ be the subtree rooted at the $j^{\mathrm{th}}$ child of $r$
9:     for $i = 1$ to $\kappa$ do
10:        Candidates $\leftarrow \varnothing$
11:        for each child $j$ of $r$ s.t. no sub-family of $T_j$ has yet been added to $\mathcal{A}^{i-1}_T$ do
12:            Add the output $\mathcal{A}^{i,j}_T$ of $D^{\mathcal{T}}(\sigma, T_j, \delta_{\mathrm{OPT}}, R')$ to Candidates
13:        Let $\mathcal{A}^{i,j*}_T$ be the tree $t$ in Candidates that maximizes $\sigma|\mathrm{cov}(t, R')| - \delta_{\mathrm{OPT}}|t|$
14:        $\mathcal{A}^i_T \leftarrow \mathcal{A}^{i-1}_T \cup \{\mathcal{A}^{i,j*}_T\}$            ▷ By construction, $\mathcal{A}^i_T$ is a tree.
15:        Add $\mathcal{A}^i_T$ to Solutions
16:        $R' \leftarrow R' \setminus \mathrm{cov}(\mathcal{A}^i_T)$
17:    return $\mathcal{A}_T$, the tree $t$ in Solutions that maximizes $\sigma|\mathrm{cov}(t, R'_0)| - \delta_{\mathrm{OPT}}|t|$
```

sharing T's root. Let $M(T, R')$ be the $t \sqsubset T$ that maximizes $|\mathrm{cov}(t, R')| - \delta_{\mathrm{OPT}}|t|$. Given tree T of height at most $\sigma + 1$, we show inductively that $D^{\mathcal{T}}$ returns some $\mathcal{A}_T \sqsubset T$ with

$$(\sigma+1)|\mathrm{cov}(\mathcal{A}_T, R')| - \delta_{\mathrm{OPT}}|\mathcal{A}_T| \geq |\mathrm{cov}(M(T,R'), R')| - \delta_{\mathrm{OPT}}|M(T,R')| \,. \qquad (2)$$

Broadly, algorithm $D^{\mathcal{T}}$ behaves as follows. If the root r of tree T has κ children, $D^{\mathcal{T}}$ makes a sequence of κ recursive calls to itself. After the $i-1^{\mathrm{th}}$ call, it has a putative solution $\mathcal{A}^{i-1}_T$ comprising r itself and $i-1$ subtrees, each hanging from a different child of r. In the i^{th} iteration, $D^{\mathcal{T}}$ adds a subtree from a "new" child to $\mathcal{A}^{i-1}_T$. This new subtree has the maximum value of $\sigma\mathrm{cov}(t, R'_i) - \delta_{\mathrm{OPT}}|t|$, where R'_i is set R' during the i^{th} iteration (before step 16). With first parameter $\sigma+1$, $D^{\mathcal{T}}$ returns the best of the $\kappa+2$ putative solutions (including $\varnothing$ and $\{r\}$), the subtree $\mathcal{A}_T$ maximizing $\sigma|\mathrm{cov}(\mathcal{A}_T, R')| - \delta_{\mathrm{OPT}}|\mathcal{A}_T|$.

▶ **Lemma 4.** *Given tree T of height $\leq \sigma+1$, $D^{\mathcal{T}}(\sigma+1, T, \delta_{OPT}, R')$, returns $\mathcal{A}_T \sqsubset T$ satisfying inequality* (2).

Proof. First, the base case. If $\sigma = 0$, and the tree has height 1, the only options are $\varnothing$ and $\{r\}$. These are easy to evaluate and inequality (2) is easily satisfied.

If $\sigma > 0$, consider tree $M(T, R'_0)$, where R'_0 is the initial *value* of R' in $D^{\mathcal{T}}$. Again, if $M(T, R'_0)$ is empty, or if it is $\{r\}$, $D^{\mathcal{T}}$ will consider those two solutions, so will return some solution satisfying (2). Therefore, assume tree $M(T, R'_0)$ comprises root r and subtrees hanging from $\kappa^* \leq \kappa$ children. We focus analysis on $\mathcal{A}^{\kappa^*}_T$. Although $D^{\mathcal{T}}$ does not know κ^*, it generates $\mathcal{A}^i_T$ for all $i \leq \kappa$, returning the *best* of these, at least *as good* as $\mathcal{A}^{\kappa^*}_T$.

We renumber root r's children to match the order in which they contribute to $\mathcal{A}^{\kappa^*}_T$, so that $j^* = i$ on each iteration. The construction of $M(T, R'_0)$ can also be interpreted iteratively, so that at iteration i it adds a subtree hanging from child number i_M; let that subtree be called $M_{i_M}(T, R'_0)$. However, we insist that if $M(T, R'_0)$ contains a subtree hanging from child $i \leq \kappa^*$, it is *chosen* at iteration i.

Consider the subtree added to $\mathcal{A}^{\kappa^*}_T$ in iteration i, $\mathcal{A}^{i,i}_T$. If $\mathcal{A}^{i,i}_T$ is different from $M_{i_M}(T, R'_0)$, it must be because $D^{\mathcal{T}}(\sigma, T_i, \delta_{\mathrm{OPT}}, R'_i)$ returned $\mathcal{A}^{i,i}_T$, which had the largest value for $t \in$

Candidates (children numbered i and above) of $\sigma|\mathrm{cov}(t, R'_i)| - \delta_{\mathrm{OPT}}|t|$, while tree $M(T, R'_0)$ added a subtree from child $i_M \geq i$, giving inequality (3), as follows

$$\begin{aligned}\sigma|\mathrm{cov}(\mathcal{A}_T^{i,i}, R'_i)| - \delta_{\mathrm{OPT}}|\mathcal{A}_T^{i,i}| &\geq \sigma|\mathrm{cov}(\mathcal{A}_T^{i,i_M}, R'_i)| - \delta_{\mathrm{OPT}}|\mathcal{A}_T^{i,i_M}| &(3)\\ &\geq |\mathrm{cov}(M(T_{i_M}, R'_i), R'_i)| - \delta_{\mathrm{OPT}}|M(T_{i_M}, R'_i)| &(4)\\ &\geq |\mathrm{cov}(M_{i_M}(T, R'_0), R'_i)| - \delta_{\mathrm{OPT}}|M_{i_M}(T, R'_0)|\,, &(5)\end{aligned}$$

while inequality (4) arises from the inductive argument about $D^{\mathcal{T}}(\sigma, \ldots)$, while inequality (5) flows from the optimality of $M(T_{i_M}, R'_i)$ on (T_{i_M}, R'_i). If in fact $\mathcal{A}_T^{i,i}$ is the same as $M_{i_M}(T, R'_0)$, then the *overall* inequality (3) – (5) holds because $\sigma \geq 1$.

The coverage of $\mathcal{A}_T^{\kappa^*}$ on R'_0 is the union of $\mathrm{cov}(\{r\}, R'_0)$ and $\cup_{i=1}^{\kappa^*}\mathrm{cov}(\mathcal{A}_T^{i,i}, R'_i)$. From the definition of R'_i (step 16 of $D^{\mathcal{T}}$), the $\mathrm{cov}(\cdot)$ sets are disjoint. Since also $\sigma \geq 1$,

$$\begin{aligned}&\sigma|\mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)| - \delta_{\mathrm{OPT}}|\mathcal{A}_T^{\kappa^*}|\\ &\geq [|\mathrm{cov}(\{r\}, R'_0)| - \delta_{\mathrm{OPT}}] + \sum_{i=1}^{\kappa^*}\left[\sigma|\mathrm{cov}(\mathcal{A}_T^{i,i}, R'_0)| - \delta_{\mathrm{OPT}}|\mathcal{A}_T^{i,i}|\right]\,,\end{aligned}$$

and, by applying *overall* inequality (3) – (5), this is

$$\begin{aligned}&\geq [|\mathrm{cov}(\{r\}, R'_0)| - \delta_{\mathrm{OPT}}] + \sum_{i=1}^{\kappa^*}[|\mathrm{cov}(M_{i_M}(T, R'_0), R'_i)| - \delta_{\mathrm{OPT}}|M_{i_M}(T, R'_0)|]\\ &= |\mathrm{cov}(\{r\}, R'_0)| + \left[\sum_{i=1}^{\kappa^*}|\mathrm{cov}(M_{i_M}(T, R'_0), R'_i)|\right] - \delta_{\mathrm{OPT}}|M(T, R'_0)|\,, \qquad (6)\end{aligned}$$

For each iteration i, inspired by Chekuri and Kumar [6], we have

$$\mathrm{cov}(M_{i_M}(T, R'_0), R'_i) \supseteq \mathrm{cov}(M_{i_M}(T, R'_0), R'_0 \setminus \mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)\,)\,,$$

$$\text{so,}\quad \mathrm{cov}(\{r\}, R'_0) \cup \bigcup_{i=1}^{\kappa^*}\mathrm{cov}(M_{i_M}(T, R'_0), R'_i) \supseteq \left(\mathrm{cov}(\{r\}, R'_0) \cup \bigcup_{i=1}^{\kappa^*}\mathrm{cov}(M_{i_M}(T, R'_0), R'_0)\right) \setminus \mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)\,.$$

Applying the union bound to the LHS and the composition of $M(T, R'_0)$ to the RHS,

$$\begin{aligned}|\mathrm{cov}(\{r\}, R'_0)| + \sum_{i=1}^{\kappa^*}|\mathrm{cov}(M_{i_M}(T, R'_0), R'_i)| &\geq |\mathrm{cov}(M(T, R'_0), R'_0) \setminus \mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)|\\ &\geq |\mathrm{cov}(M(T, R'_0), R'_0)| - |\mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)|\,.\end{aligned}$$

Combining this with the inequality ending at (6), we see that

$$\begin{aligned}\sigma|\mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)| - \delta_{\mathrm{OPT}}|\mathcal{A}_T^{\kappa^*}| &\geq \Big(|\mathrm{cov}(M(T, R'_0), R'_0)| - |\mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)|\Big) -\\ &\qquad \delta_{\mathrm{OPT}}|M(T, R'_0)|\,,\\ \therefore\quad (\sigma+1)|\mathrm{cov}(\mathcal{A}_T^{\kappa^*}, R'_0)| - \delta_{\mathrm{OPT}}|\mathcal{A}_T^{\kappa^*}| &\geq |\mathrm{cov}(M(T, R'_0), R'_0)| - \delta_{\mathrm{OPT}}|M(T, R'_0)|\,. \qquad \blacktriangleleft\end{aligned}$$

For a forest of out-trees, we return the best individual tree solution generated by $D^{\mathcal{T}}$. Without knowing the value of δ_{OPT}, there are only mn possible values, so we can in polynomial time try them all and return the densest sub-forest. Finally, for OPT, the right-hand side of inequality (2) is zero, so Lemma 4 shows $D^{\mathcal{T}}$returns a tree with density at least δ_{OPT} divided by maximum tree height.

4 Hardness of approximation

Apart from the in-tree case, which is in P, the approximation *factors* for PCMSSC are polynomial. In this section, we show that such factors are to be "expected". We show a hardness *reduction* to PCMSSC from the PLANTED DENSE SUBGRAPH CONJECTURE, which is a statement about the difficulty of finding a dense component in an Erdős-Rényi graph [5]. Our inspiration here is Burge, Munagala and Srivastava's hardness-of-approximation reduction from DENSEST k-SUBGRAPH for pipelined query operators [4].

First, we define the PLANTED DENSE SUBGRAPH CONJECTURE, where N is the input graph's order, α its log density, k the order of the planted component, and β its log density.

▶ **Definition 5.** The problem PDS(N, k, α, β) has as input a graph with probability 1/2 drawn from $G(N, N^{\alpha-1})$ and with probability 1/2 drawn from $G(N, N^{\alpha-1})$, in which some k vertices are chosen uniformly from N and on them is added a subgraph drawn from $G(k, k^{\beta-1})$. The task is to correctly report from which of the two distributions the graph was drawn.

▶ **Conjecture 6** (PLANTED DENSE SUBGRAPH CONJECTURE). *For all $\varepsilon > 0$, $k \geq \sqrt{N}$, and $\beta < \alpha$, no probabilistic polytime algorithm can, with advantage $> \varepsilon$, solve* PDS(N, k, α, β).

We show that identifying an ordering π with low PCMSSC score solves PDS almost surely.

▶ **Lemma 7.** *Assuming the* PLANTED DENSE SUBGRAPH CONJECTURE, *there is no poly-time algorithm that, for $\varepsilon > 0$, approximates* PCMSSC *within factor $O(n^{1/6-\varepsilon})$ nor $O(m^{1/12-\varepsilon})$.*

Proof. First, let $k = \sqrt{N}$, $\alpha = 1/2$ and $\beta = 1/2 - \gamma$, for some $\gamma > 0$. Given graph $H([N], \mathcal{E})$, the input to PDS, define family $\mathcal{F}$ to be $N\lambda$ *vertex sets*, together with an *edge set* for each edge in $\mathcal{E}$. More specifically, the vertex sets are $V_{u,i}$ for each $u \in \{1, \ldots, N\}$ and $i \in \{1, \ldots, \lambda\}$, while the edge sets are $E_{u,v}$ for each $(u, v) \in \mathcal{E}$. For convenience, let $U = \{0, 1, \ldots, n\}$, so that $|U| = n + 1$, with $U^+ = \{1, \ldots, n\}$.

The $\prec$ relation acts as follows: every edge set must be preceded by all "copies" of each of its endpoints' vertex sets. That is, for all $(u, v) \in \mathcal{E}$ and for all i, $V_{u,i} \prec E_{u,v}$ and $V_{v,i} \prec E_{u,v}$.

We now define the composition of the sets in $\mathcal{F}$. Every vertex set comprises the same item: $V_{u,i} = \{0\}$, for all u, i. To define the edge sets, we *associate* items with vertices: let the set of items associated with vertex v be U_v. (This association is merely a vehicle to define edge sets, and is distinct from vertex-set composition.) The construction is randomized, based on parameter $p \in (0, 1)$: each item in U^+ is associated with each vertex in H independently, with probability p. Hence $\mathbf{E}[|U_v|] = np$, while for all $j \in U^+$, $\mathbf{E}[|\{v : j \in U_v\}|] = Np$. For each edge (u, v), we define $E_{u,v}$ to be $U_u \cap U_v$, with expected size np^2.

Choosing p. If there is a planted component, $\mathcal{P}$, it (just) covers all of U: this drives our selection of p. There are $\sqrt{N}$ vertices in $\mathcal{P}$, and each item j is in $\sqrt{N}p$ of the U_v sets, on average. We expect (ignoring constants) around Np^2 vertex pairs in $\mathcal{P}$ where j is associated with both. The probability of each vertex pair having an edge is, independently, $k^{\beta-1} = (N^{1/2})^{(-1/2-\gamma)} = N^{-(1/4+\gamma/2)}$. Therefore, $\mathbf{E}[|E_{u,v} \in \mathcal{P} : j \in E_{u,v}|] \approx Np^2 N^{-(1/4+\gamma/2)} = N^{3/4-\gamma/2}p^2$. So this is close to 1, we set $p = 32 \cdot N^{(-3+\gamma')/8} \log N$, with $\gamma' = 2\gamma$.

Planted component. If there is a planted component $\mathcal{P}$, we show a "good" PCMSSC solution. Suppose that π starts with all $V_{u,i}$ for all $u \in \mathcal{P}$ followed by $E_{u,v}$ for all $u, v \in \mathcal{P}$ (edges $\mathcal{E}$ in $\mathcal{P}$). The number of sets so far is $\lambda\sqrt{N}$ plus a random value highly concentrated around $\sqrt{N}(\sqrt{N} - 1)N^{-(1/4+\gamma/2)}/2$, hence $\leq N^{3/4-\gamma/2}$ with high probability (whp). The claim below proves U is covered whp by these $\mathcal{P}$-based sets. If so, the PCMSSC cost is whp bounded by $n(\lambda\sqrt{N} + N^{3/4-\gamma/2})$.

▶ **Claim 8.** *If H was generated with a planted component, $\mathcal{P}$, then with high probability all items in U^+ are covered by the planted component's edge sets.*

Proof of Claim 8. The expected number of U_v in $\mathcal{P}$ containing j is $\sqrt{N}p$, so

$$\Pr[|v \in \mathcal{P} : j \in U_v| \leq \sqrt{N}p/2] \leq \exp(-\sqrt{N}p/8) = \exp(-4 \cdot N^{(1+\gamma')/8} \log N)\,,$$

via Chernoff bounds, which is tiny. Therefore, and since $\binom{x}{2} \geq x^2/4$ for large enough x, the probability that the number of pairs of vertices in $\mathcal{P}$ where j is associated with both is at most $Np^2/16$ is (also) at most $\exp(-4 \cdot N^{(1+\gamma')/8} \log N)$.

For each such pair, there is an edge actually present with probability $N^{-(1/4+\gamma/2)}$. Item-association and edge-presence events are independent, so in expectation j is in at least

$$\mu = (Np^2/16)N^{-(1/4+\gamma/2)} = 32^2(\log^2 N)/16$$

edge sets in $\mathcal{P}$. For all $j \in U^+$, again via Chernoff bounds,

$$\Pr[|\{(u,v) \in \mathcal{E}_{\mathcal{P}} : j \in E_{u,v}\}| \leq \mu/2] \leq \exp(-(64 \log N)/8) = N^{-8}\,.$$

Taking the union over all n items in U^+, the probability that some item is uncovered by the edge sets in $\mathcal{P}$ is at most n/N^8, which is very small (below, we choose n to be $o(N)$). ◀

No planted component. We show that if there is no planted component, even after several vertex and edge sets have appeared in π, there is a significant portion of items not yet covered, pushing the PCMSSC score very high.

Consider a solution to PCMSSC derived from x vertices and their induced edges. The number of vertex sets is λx and the number of vertex pairs is $\approx x^2/2$. The expected number of edges is $\leq x^2/\sqrt{N}$, but there are in fact $\binom{N}{x} \leq N^x$ such sets of N vertices. Via the Chernoff bound, the probability that all of them have at most 3/2 their average number of edges, $\mu_{\mathcal{E}}$, is at most $N^x \exp(-\mu_{\mathcal{E}}/12)$. If we would like this probability to be at most $1/N^8$, say, then let $\mu_{\mathcal{E}} \geq 12(x+8)\log N$. Since the number of vertex pairs is at least $x^2/4$, and hence $\mu_{\mathcal{E}} \geq x^2/(4\sqrt{N})$, there is a very low probability of exceeding $x^2/\sqrt{N}$ edges across all sets of x vertices if $x \geq 500\sqrt{N}\log N$.

By construction, each edge set has on average np^2 items. The probability that some edge set related to at least one of the at most $x^2/\sqrt{N}$ edges has more than $2np^2$ items is at most $(x^2/\sqrt{N})\exp(-np^2/12)$. If $n \geq 200 \log N/p^2$, which is satisfied by $n \in \Omega(N^{3/4})$, because $x \leq N$, this probability is tiny.

Even ignoring the possibility of edge sets covering common items, we conclude that whp, after λx vertex sets and $x^2/\sqrt{N}$ edge sets appear in π, at most $2np^2x^2/\sqrt{N} \leq 2048 \cdot nN^{(-5+\gamma')/4}x^2(\log^2 N)$ items have been covered. Hence if $x \leq N^{(5-2\gamma')/8}$, then there are $o(n)$ items covered. With $\Omega(n)$ items uncovered, the PCMSSC score is at least a constant multiplied by $n(\lambda N^{(5-2\gamma')/8} + N^{(3-2\gamma')/4})$.

Letting λ grow to at least $N^{1/4}$ and letting γ' shrink, whp the asymptotic ratio between this cost and the cost in the planted case (just before Claim 8) tends arbitrarily close to $N^{1/8}$. Since $n \in \Theta(N^{3/4})$ permits our probability bounds, whp there is a gap of $\Theta(n^{1/6-\varepsilon})$, for all $\varepsilon > 0$, in the PCMSSC costs between the two cases. Likewise, the number of sets m in the input is highly concentrated around $\lambda N + N^{3/2}$, which is $\Theta(N^{3/2})$. An algorithm for PCMSSC with an approximation factor better than $\Theta(n^{1/6-\varepsilon})$ or $\Theta(m^{1/12-\varepsilon})$, for all $\varepsilon > 0$, could solve PDS with very significant advantage. From this, we conclude Lemma 7. ◀

5 Related work

The Min-Sum Vertex Cover problem – the special case of MSSC where each set has two items – admits a 2-approximation algorithm [10]. In database theory, MSSC has been referred to as the Pipelined Set Cover problem. This model typically allows for different processing times (or costs) on the sets, and has been given an alternative (but still factor-4) approximation algorithm [19].

In web search theory, a generalization of MSSC relating the min-sum objective to Minimum Latency Set Cover is called Multiple Intents Reranking. This extends the objective function to allow for different *user profiles* – relating to how many times each item needs to be covered [1]. It has also been referred to as the Generalized Min-Sum Set Cover problem, which has constant-factor approximations [2].

In Scheduling with AND/OR Precedence Constraints (SAOPC), there are two types of jobs: AND-jobs are available only when all precedences are met, while OR-jobs only require that at least one of the precedences are met [11]. In fact, PCMSSC is a special case of SAOPC in which the precedence-constrained sets are AND-jobs, and the item are OR-jobs. Scheduling to minimize the makespan is Label Cover-hard for general AND/OR precedences [13], but the reduction has an OR-AND-OR-AND structure: it is unclear whether PCMSSC is Label Cover-hard, that is [8], hard to approximate within $2^{\log^{1-1/(\log\log^c n)} n}$.

Erlebach, Kääb, and Möhring prove that minimizing the total completion time in SAOPC is Label Cover-hard, but that scheduling available jobs with Smith's shortest processing time (SPT) rule gives an $O(n)$-approximation [9]. They further prove that this algorithm gives an $O(\sqrt{n})$-approximation for the special case with a single processor where all jobs have equal weights. However, in PCMSSC, AND jobs have zero weight and unit processing time, but OR jobs have unit weight and zero processing time.

Finally, in the Partially Ordered Knapsack (aka POK), there is a predecessor $\prec$ relation, inducing a directed graph, and each vertex has a profit and a weight [15]. The aim is to find a *closed under predecessor* set of vertices that maximizes total profit subject to a total weight constraint. For in- and out-trees, Johnson and Niemi consider pseudo-polynomial algorithms that are nonetheless efficient in practice, as well as an FPTAS (with running time proportional to $1/\varepsilon$) derived from a standard PTAS approach for knapsack problems [15].

Kolliopoulos and Steiner [16] provide an FPTAS for POK when the underlying order is two dimensional. They note that Hajiaghayi et al. [14] show that POK is hard to approximate within $2^{\log^\delta n}$, that POK generalizes Densest k-Subgraph, and that a constant-factor algorithm for Partially Ordered Knapsack would provide a sub-factor-2 approximation for PCSTW. Indeed, the Minimum-Rank Precedence-Closed Subgraph problem that Chekuri and Motwani solve optimally is a minimize-ratio version of POK; again, there is some connection to the difference in approximability between Densest Subgraph and Densest k-Subgraph. Borradaile, Heeringa, and Wilfong [3] examine variants of POK with undirected graphs, and 1-neighbor as well as all-neighbor precedence requirements.

Acknowledgements. We thank Tim Miller for the foundational software-testing problem.

References

1 Yossi Azar, Iftah Gamzu, and Xiaoxin Yin. Multiple intents re-ranking. In *STOC: 41st ACM Symposium on Theory of Computing*, pages 669–678, 2009.

2 Nikhil Bansal, Anupam Gupta, and Ravishankar Krishnaswamy. A constant factor approximation algorithm for generalized min-sum set cover. In *SODA: 21st ACM-SIAM Symposium on Discrete Algorithms*, pages 1539–1545, 2010.

3 Glencora Borradaile, Brent Heeringa, and Gordon Wilfong. The knapsack problem with neighbour constraints. *Journal of Discrete Algorithms*, 16:224–235, 2012.

4 Jen Burge, Kamesh Munagala, and Utkarsh Srivastava. Ordering pipelined query operators with precedence constraints. Technical Report 2005-40, Stanford InfoLab, 2005.

5 Moses Charikar, Yonatan Naamad, and Anthony Wirth. On approximating target set selection. In *APPROX: 19th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 4:1–4:16, 2016.

6 Chandra Chekuri and Amit Kumar. Maximum coverage problem with group budget constraints and applications. In *APPROX: 7th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems*, pages 72–83, 2004.

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

8 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Information Processing Letters*, 89(5):247–254, 2004.

9 Thomas Erlebach, Vanessa Kääb, and Rolf H Möhring. Scheduling AND/OR-networks on identical parallel machines. In *WOAO: International Workshop on Approximation and Online Algorithms*, pages 123–136, 2003.

10 Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004.

11 Donald W. Gillies and Jane W.-S. Liu. Scheduling tasks with and/or precedence constraints. *SIAM Journal on Computing*, 24(4):797–810, 1995.

12 A. V. Goldberg. Finding a maximum density subgraph. Technical Report CSD-84-171, UC Berkeley Computer Science Division, 1984.

13 Michael Goldwasser and Rajeev Motwani. Intractability of assembly sequencing: Unit disks in the plane. In *WADS: Workshop on Algorithms and Data Structures*, pages 307–320, 1997.

14 M Hajiaghayi, Kamal Jain, L Lau, I Măndoiu, Alexander Russell, and V Vazirani. Minimum multicolored subgraph problem in multiplex pcr primer set selection and population haplotyping. *Computational Science–ICCS 2006*, pages 758–766, 2006.

15 David S. Johnson and K. A. Niemi. On knapsacks, partitions, and a new dynamic programming technique for trees. *Mathematics of Operations Research*, 8(1):1–14, 1983.

16 Stavros G. Kolliopoulos and George Steiner. Partially ordered knapsack and applications to scheduling. *Discrete Applied Mathematics*, 155(8):889–897, 2007.

17 Jesssica McClintock, Tim Miller, and Anthony Wirth. Prioritisation of test suites containing precedence constraints, 2017. submitted.

18 Tim Miller and Shifa-e-Zehra Haidry. Using dependency structures for prioritization of functional test suites. *IEEE Transactions on Software Engineering*, 39(2):258–275, 2013.

19 Kamesh Munagala, Shivnath Babu, Rajeev Motwani, and Jennifer Widom. The pipelined set cover problem. In *ICDT: 10th International Conference on Database Theory*, pages 83–98, 2005.