Active Learning and Covering Problems with Precedence

author names withheld

Editor: Under Review for COLT 2026

Abstract

In the Bayesian Active Learning a hidden hypothesis is required to be uncovered. To do so, the learner is allowed to perform tests, each of which reveals partial information about the hidden hypothesis. Upon receiving this information, the learner adaptively selects the next test to be performed. The goal is to uncover the hidden hypothesis while performing as few tests as possible in the worst or average case.

In the covering problems, we are given a set of items and a collection of subsets that cover these items. The objective is to select a sequence of subsets that covers all items, which minimizing the worst or average covering cost.

For both types of problems, a natural constraint may arise that some tests can only be performed only after certain other tests (or some subsets can only be selected after selecting certain other subsets). We model such constraints using directed acyclic graphs (DAGs) that impose precedence on the tests or subsets. This paper explores the connection of active learning and covering problems under such constraints.

We show that given any bicriteria $(O(1)\,,\alpha)$-approximation ratio for the Precedence Constrained Set Cover, we can obtain an $O(\alpha \cdot \log n)$-approximation ratio for the Worst Case Active Learning with precedence constraints, where n is the number of hypothesis. Similarly, we prove that given any $O(\beta)$-approximation ratio for the Precedence Constrained Min-Sum Set Cover, we can obtain an $O(\beta \cdot \log n)$-approximation ratio for the Average Case Active Learning with Precedence Constraints. Finally, we provide several approximation algorithms for the Set Cover and Min-Sum Set Cover problems with various types of precedence constraints.

Keywords: Bayesian active learning, Set cover, Precedence constraints, Approximation Algorithms, Decision Trees

Ogolne uwagi:

- **wszelkie uwagi pisze jako komenda ... dzieki czemu na koniec latwo sie pozbyc; nie bede raczej uwag dorzucal w mailach, aby nie zniknelo; wszystko co ponizej oczywiscie do dyskusji, a gdy bedzie zgoda i bedzie zaimplementowane, to bede usuwal artefakty.**

- **przejrzawszy sporo papierow z poprzedniego COLT, mam obserwacje, ze dobrym/typowym ukladem papieru jest: intro; our contribution; related work; outline(opcjonalnie); preliminaries; wyniki; appendix.**

- **intro zwykle nie jest zbyt rozlekle oraz prawie zawsze pozbawione lania wody. Czesto od razu definicja problemu, aby wprowadzic pojecia, aby moc szybko formalnie podac wyniki (our contribution)**

- **“front” artykulu, czyli wszystko do preliminaries to typowo 4-5 stron,**

- **front we wszystkich miejcach zawiera zwykle odnosniki do literatury a sekcja “related work” jest czesto tytulowana “other related work” lub cos w tym rodzaju**

© 2026 .

- **czytelnik powiniem poza dowodami rozumiec baze przeczytawszy front (czyli rozumiec wyniki, widziec co papier robi) a jesli chce sie dowiedziec jak/dlaczego (dowody) to idzie dalej. Czesto recenzent jest leniwy i nie zajrzy dalej niz front... niestety.**
- **W zwiazku z powyzszym sekcja “our contribution” (potencjalnie w tytule dodamy “and techniques” jak Michal sugeruje) powinna sie pochwalic takze jakimis ciekawszymi trickami lub technikami uzytymi pozniej w dowodach.**

1. Introduction

pomysl na intro: zdefiniowac dwa glowne problemy: learning + set cover; zapowiedziec, ze sa powiazane ze soba i celem papieru jest przestudiowanie tych zaleznosci plus usyskanie konkretnych wynikow; pytanie/do sprawdzenia: czy ktores wyniki przypadkiem poprawiaja lub sa tozsame z najlepszymi znanymi bez precedensow; ewentualne inna "marketingowe" uwagi.

Consider a set $\mathcal{H}$ of n *hypotheses*, a set $\mathcal{T}$ of m *tests* and an unknown *target hypothesis* $h^* \in \mathcal{H}$ that needs to be discovered through an adaptive learning process. Each test $t \in \mathcal{T}$ is a partition of $\mathcal{H}$, that is, t consists of subsets of $\mathcal{H}$ such that $x \cap y = \emptyset$ for any $x.y \in t$ and $\bigcup t = \mathcal{H}$. As a result of executing a test $t \in \mathcal{T}$, questioner recives a *reply* that reveals $x \in t$ such that $h^* \in x$. That is, the questioner learns which subset of $\mathcal{H}$ that belongs to t contains the target. Each subsequent test is selected by questioner by taking into acocunt replies from all test to date. Without formally stating an optimization criterion we refer to the above as the *Adaptive Learing Process* (AL). (Another widely used name in the literature is the decision tree construction). The goal for the questioner is to output h^*.

Consider an arbitrary partial order $(\mathcal{T}, \preceq)$ that introduces a precedence relation between tests. This leads us to the two adaptive learning problems in which order to perform a test t, all its predecessors had to be performed previously. Hence we have the *Worst Case Adaptive Learning with Precedences* (PCWCAL) in which the goal is to compute the AL that respects the precedence constraints and outputs the target h^* by performing the minimum number of tests in the worst case. Similarly, in the *Average Case Adaptive Learning with Precedences* (PCACAL) the optimization criterion changes to minimizing the number of queries done on average.

In this work we study connections between adaptive lerning with precedences and the covering problems defined as follows. We are given a set $\mathcal{U}$ of n items, a collection $\mathcal{S}$ of m subsets of $\mathcal{U}$, such that $\bigcup \mathcal{S} = \mathcal{U}$, an arbitrary partial order $(\mathcal{S}, \preceq)$ on these subsets and an integer k. We say that a subfamily $\mathcal{C} \subseteq \mathcal{S}$ *covers* at least k items from $\mathcal{U}$ if $|\bigcup \mathcal{C}| \geq k$. We ask for a $\mathcal{C} \subseteq \mathcal{S}$ that covers at least k items from $\mathcal{U}$ and for each $x \in \mathcal{C}$ and each $y \in \mathcal{S}$ such that $y \preceq x$ it holds $y \in \mathcal{C}$. In the *Precedence Constrained Set Cover* (PCSC) the goal is to minimize $|\mathcal{C}|$. A permutation $C_1, \ldots, C_k)$ of the elements in $\mathcal{C}$ is *consistent* with the partial order $(\mathcal{S}, \preceq)$ if for any C_i and C_j such that $C_i \preceq C_j$ it holds $i < j$. The *coverage time* of a $x \in \bigcup \mathcal{C}$ is the minimum index i such that $x \in C_i$. In the *Precedence Constrained Min-Sum Set Cover* (PCMSSC) the goal is to find a sequence $(C_1, \ldots, C_k)$ that minimizes the total coverage time of all items in $C_1 \cup \cdots \cup C_k$.

1.1. Our contribution

Pomysl na rozdzial:

- **zajawka, że wprpwadzimy nowe inne problemy (raz - jak pomocnicze; dwa - jako dopelnienie obrazu roznych rzeczy z litearatury)**
- **zdefiniowac pozostałe problemy z "diagramu" zaleznosci miedzy nimi**
- **diagram**
- **najwazniejsze twierdzenia**
- **tabelka na podsumowanie**

Consider following problems:

- The *Precedence Constrained Bayesian Active Learning Problem* consists a set of $\mathcal{H}$ of n hypothesis, a set $\mathcal{T}$ of m tests and a DAG (directed acyclic graph) $\mathcal{F} = \{\mathcal{T}, \preceq\}$ encoding the precedence constraints between available tests. Among $\mathcal{H}$ a hidden hypothesis is required to be encovered. To do so, the learner is allowed to perform tests, each of which reveals partial information about the hidden hypothesis. Upon receiving this information, the learner adaptively selects the next test to be performed. Importantly, in order to perform such test the learner needs to perform all of its predecesors in $\mathcal{F}$ first. The goal is to uncover the hidden hypothesis while performing as few tests as possible. Depending on the chosen criterion we distinguish between the *Precedence Constrained Worst Case Active Learning* (PCWCAL) and *Precedence Constrained Average Case Active Learning* (PCACAL) problems.

- The *Precedence Constrained Covering Problem* consists of a set of n items $\mathcal{U}$, a collection $\mathcal{S}$ of m subsets of $\mathcal{U}$ that cover these items, and a DAG $\mathcal{F} = \{\mathcal{S}, \preceq\}$ encoding the precedence constraints between available subsets. The goal is to select a sequence of tests that covers at least K items. Depending on the chosen criterion we distinguish between the *Precedence Constrained Set Cover* (PCSC) and *Precedence Constrained Min-Sum Set Cover* (PCMSSC) problems. In the first we are only interested in minimizing the number of selected subsets, while in the second we want to minimize the average time it takes to cover an item.

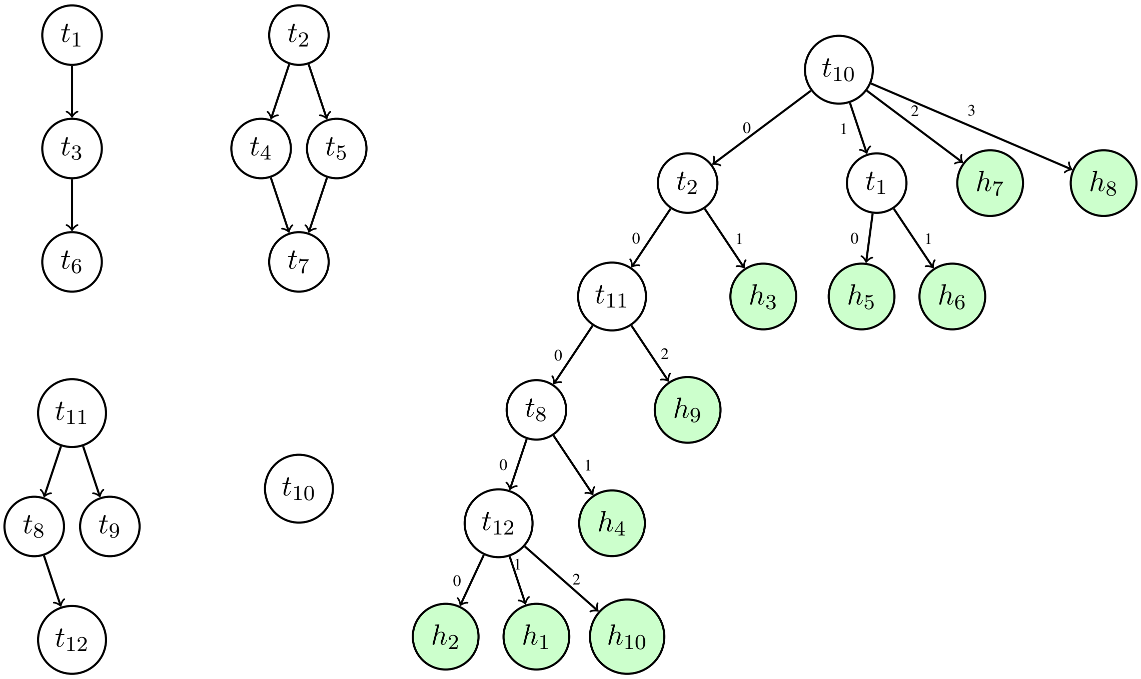
1.2. Our results and techniques

precedence/problem	**PCSC**	**PCMSSC**	**PCWCAL**	**PCACAL**
none	$O(\log n)$	4	$O(\log n)$	$O(\log n)$
inforest	$O(\log n)$*	4	$O(\log n)$*	$O(\log n)$*
outforest	$O(\log^2 n)$**	$O(\log n)$**	$O(\log^2 n)$*	$O(\log^2 n)$*
general	$O(\sqrt{m} \log n)$*	$O(\sqrt{m})$	$O(\sqrt{m} \log n)$*	$O(\sqrt{m} \log n)$*

Table 1: Approximation algorithms for various covering and active learning problems under different precedence constraints. (* denotes new results, ** denotes previously unmentioned corollaries of known results).

	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
h_1	0	0	0	0	0	0	2	0	0	0	0	1
h_2	0	0	0	0	0	1	0	0	0	0	0	0
h_3	0	1	0	1	0	0	0	0	0	0	0	0
h_4	0	0	0	0	1	0	0	1	0	0	0	0
h_5	0	0	0	0	0	0	1	0	0	1	1	0
h_6	1	0	0	2	0	0	0	0	0	1	0	0
h_7	0	0	0	0	0	2	0	0	1	2	0	0
h_8	0	0	1	0	0	0	0	0	0	3	0	0
h_9	0	0	0	0	0	0	0	1	1	0	2	0
h_{10}	0	0	0	1	0	0	0	0	0	0	0	2

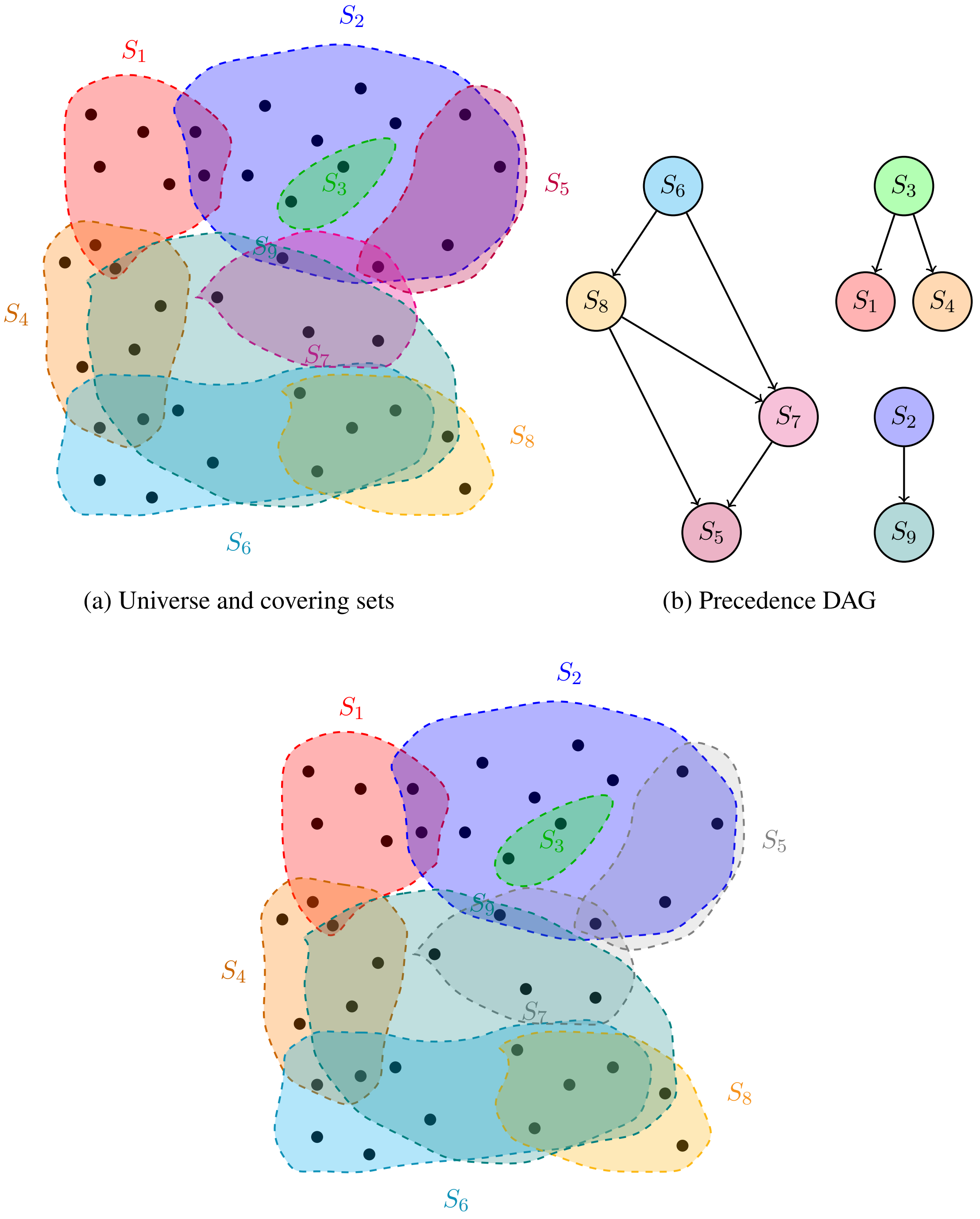
(a) Hypotheses and tests table



(b) Precedence DAG

(c) Valid decision tree

Figure 1: Example of a PCAL instance with 10 hypotheses and 12 tests. (a) Hypotheses-tests table. (b) Precedence DAG with four components. (c) A valid decision tree solution respecting precedence constraints.



(a) Universe and covering sets

(b) Precedence DAG

(c) Solution: selected sets $\{S_1, S_2, S_6, S_8, S_9\}$ (in color)

Figure 2: Example of a PCCP instance with 39 elements and 9 covering sets. (a) Universe with covering sets. (b) Precedence DAG with three components. (c) Solution using 7 selected sets (colored).

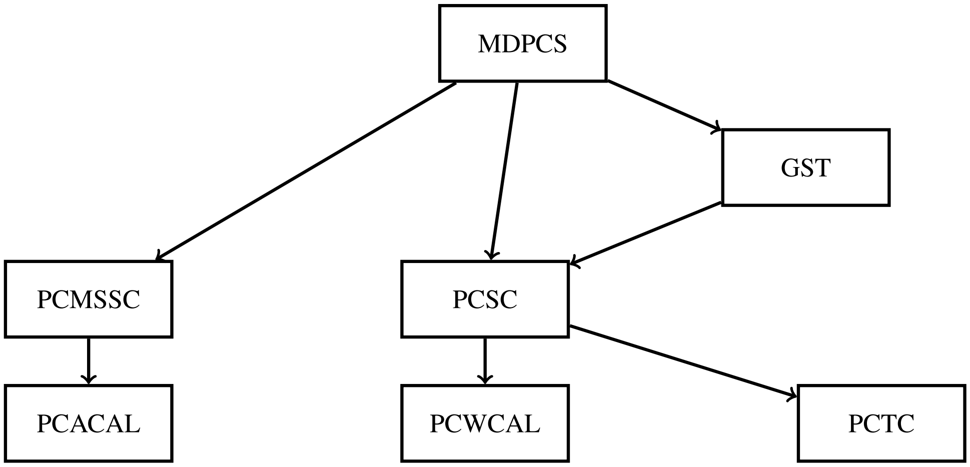


Figure 3: Relationships between covering and active learning problems, $\Pi_1 \rightarrow \Pi_2$ denotes that an approximation algorithm for problem Π_1 implies an approximation algorithm for problem Π_2.

2. Preliminaries

Propozycje do notacji:

- **odnosnie czesciowych porzadkow, to moze pozostac przy akademickim $(\mathcal{T}, \preceq)$**
- **drzewo decyzyjne to D, to niech D_v oznacza poddrzewo ukorzenione w tescie v; wowczas $V(D_v)$ donosiloby sie do wierzcholkow/testow ponizej v, wlacznie z v**
- **moze cos w rodzaju leaves(D') oraz internal(D') do oznaczanie lisci i wierzch. wewnetrznych drzewa**
- **do oznaczania sekwencji zapytan do wierzcholka v w D moznaby uzywac standardowego $D_{r\to v}$**
- **proponuje, aby wszelkie parametry byly komendami latex ($cov()$, nazwy problemow (to juz wprowadzilem).**

In all adaptive learning problems, the goal is to design a *strategy* $\mathcal{S}$ understood as an adaptive algorithm, that based on the previous responses provides the questioner with the next test to be performed. We represent this strategy as a rooted tree D called a *decision tree*, where each internal node represents a test to be performed and each leaf represents a hypothesis. Formally, a decision tree is defined recursively as follows. If $r = (R_1, \ldots, R_l) \in \mathcal{T}$ is the first test performed by $\mathcal{S}$, then r is the root of D. For each reply R_i, $i \in \{1, \ldots, l\}$, take inductively defined decision tree D_i derived from the queries done by $\mathcal{S}$ when the reply to r is R_i. Then, D is obtained by making the roots of $D_1, \ldots, D_l$ the children of r. To complete the inductive definition, whenever $\mathcal{S}$ finalizes the search by declaring that $h \in \mathcal{H}$ is the target, the corresponding decistion subtree is the vertex h. Hence, each edge outgoing from the root r of D is associated with a unique response to a test r, and the same holds for all internal It should be remarked that it is possible for a test to appear multiple times in the decision tree. Hence, with a slight abuse of notation we use tests as vertices of D but whenever a given test provides many vertices in D we make clear which one we refer to.

Let $\mathrm{tests}(D, h)$, denote the sequence of tests performed when the hidden hypothesis is h and the learner follows the strategy represented by decision tree D. $\leftarrow$ to chyba nie jest uzywane (check todo). The cost of identifying hypothesis h using a decision tree D is denoted by $\texttt{COST}(D, h)$ and defined as the number of tests performed by D when the target is h. We consider two cost measures for decision trees: the worst-case cost $\texttt{COST}_W(D) = \max_{h\in\mathcal{H}} \texttt{COST}(D, h)$ and the average-case cost $\texttt{COST}_A(D) = \sum_{h\in\mathcal{H}} \texttt{COST}(D, h)$ (up to a multiplicative factor of $1/|\mathcal{H}|$). The criteria we consider for any PCAL instance I are denoted by

$$\texttt{OPT}_A(I) = \min\{\texttt{COST}_A(D) \mid D \text{ is a decision tree}\},$$

$$\texttt{OPT}_W(I) = \min\{\texttt{COST}_W(D) \mid D \text{ is a decision tree}\}.$$

Skoro mamy już te problemy PCWCAL oraz PCACAL zdefiniowane, to tutaj aby nie powtarzac w nowej notacji drze decyzyjnych zamienie to na uwage/obserwacje w ramach jednego paragrafu do optymalizacji czego to sie sprowadza odnosnie drzew dec. Hence the PCACAL aims at finding a decision tree D such that $\texttt{COST}_A(D) = \texttt{OPT}_A(I)$ and in case of PCWCAL the goal is to find D such that $\texttt{COST}_W(D) = \texttt{OPT}_W(I)$.

For any subtree D' of D, $\mathrm{tests}(D',)$ denotes the subset of $\mathcal{T}$ that appear in D'. $\leftarrow$ to chyba nie jest uzywane (check todo) We write $\mathrm{inner}(D)$ we denote the set of nodes in D that are not leaves. Hence, each usage of a test in D corresponds to a unique element of $\mathrm{inner}(D)$. **W sumie sie wole upewnic co mamy na mysli piszac "not distinguished"? Ja to czytam jako zbior hipotez bedacych potencjalnymi targetami?** Let $\mathcal{H}_t$ denote the set of hypotheses that are not yet distinguished by the tests selected before test $t \in \mathrm{inner}(D)$ in the decision tree. **Trzebaby zrobic jaki fix na tą notację, może $\mathcal{H}(D_v)$ jako zbiór hipotez gdy dochodzimy do podrzewa D_v. Odnoszenie się do wierzchołka drzewa zamiast hipotezy ma tą zaletę, że czasem mamy warianty gdzie jedna hipoteza jest w wielu miejscach w drzewie, a wówczas ta notacja się psuje.** Then we immediately obtain the following simple observation:

Observation 1 *Let D be any decision tree for $\mathcal{I} = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. Then we have that:*

$$\mathit{COST}_A(D) = \sum_{t \in \mathrm{inner}(D)} |\mathcal{H}_t|$$

unifikacja I vs $\mathcal{I}$

We will also make use of the following folklore lemmas which are due to the fact that any decision tree for I can be restricted to a decision tree for each subproblem $(\mathcal{H}_i, \mathcal{T}, \mathcal{F})$ without increasing the cost:

Lemma 1 *Let $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ be any* PCAL *instance. Let $\mathcal{H}' \subseteq \mathcal{H}$. Then $\mathit{OPT}((\mathcal{H}', \mathcal{T}, \mathcal{F})) \leq \mathit{OPT}(I)$.*

Lemma 2 *Let $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ be an instance of* PCACAL. *Let $\mathcal{H}_1, \ldots, \mathcal{H}_t \subseteq \mathcal{H}$ such that $\bigcup_{i=1}^{t} \mathcal{H}_i \subseteq \mathcal{H}$ and for any $i \neq j$, $\mathcal{H}_i \cap \mathcal{H}_j = \emptyset$. Then we have that:*

$$\mathit{OPT}_A(I) \geq \sum_{i=1}^{t} \mathit{OPT}_A((\mathcal{H}_i, \mathcal{T}, \mathcal{F}))$$

Definition 3 (Group Steiner Tree (GST)) *Given an undirected graph $G = (V, E)$ with edge costs, a root vertex $r \in V$, and groups $g_1, \ldots, g_k \subseteq V$, find a minimum-cost tree T rooted at r that contains at least one vertex from each group g_i.*

For a subfamily $\mathcal{A} \subseteq \mathcal{F}$, we define the coverage as:

$$\mathrm{cov}(\mathcal{A}) \equiv \bigcup_{A \in \mathcal{A}} A$$

For a subset X of the universe, the coverage on X is:

$$\mathrm{cov}(\mathcal{A}, X) = \mathrm{cov}(\mathcal{A}) \cap X$$

The density Δ of a nonempty subfamily $\mathcal{A}$ on subset X is:

$$\Delta(\mathcal{A}, X) \equiv \frac{|\mathrm{cov}(\mathcal{A}, X)|}{|\mathcal{A}|}$$

For convenience, we define $\Delta(\emptyset, X) < 0$.

Definition 4 (Max-Density Precedence-Closed Subfamily (MDPCS)) *Given a family of m sets $\mathcal{G}$, a precedence relation $\prec$, and a set of n items to be covered $R \subseteq \mathrm{cov}(\mathcal{G})$, the MDPCS problem asks to find a precedence-closed subfamily $\mathcal{A} \subseteq \mathcal{G}$ that maximizes $\Delta(\mathcal{A}, R)$.*

For $S \in \mathcal{G}$, let $P[S]$ denote the minimal precedence-closed subfamily of $\mathcal{G}$ containing S (i.e., the ancestors of S including S itself).

3. A warm up: Binary search with precedences

In this section, we consider the special case of the PCWCAL and PCACAL where the instance $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ is an instance of the binary search problem with precedence constraints. In this setup we are given a linearly ordered set of n elements $\mathcal{H} = \{h_1, h_2, \ldots, h_n\}$ with $h_1 \prec h_2 \prec \ldots \prec h_n$ and a set of tests $t_{i,j} \in \mathcal{T}$ corresponding to performing a comparison operation informing the learner whether the target element is less than or equal to h_j or greater than h_j. This problem is NP-hard (see Section **??**) however it is possible to derive an $O(\log n)$-approximation algorithm for both the PCWCAL and PCACAL.

The algorithm for a worst case is simple. We use the equivalence between binary searching in an ordered set and the edge ranking coloring of a path. An edge ranking of a path is a coloring of its edges such that any path between two edges of the same color contains an edge of a lower color Lam and Yue (2001). Intuitevely, the color corresponds to the level of the decision tree where the test corresponding to the edge is performed. Let the input path be P. For any test $t \in \mathcal{T}$ define its *depth* as $d(t) = \max_{\tau \in \mathcal{T}, P_{\tau,t} \in F}\{d(\tau, t)\} + 1$. Let the height of $\mathcal{F}$ be defined as $h(\mathcal{F}) = \max_{t \in \mathcal{T}}\{d(t)\}$. The algorithm starts by partitioning $\mathcal{T}$ into $h(\mathcal{F})$ sets $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_{h(\mathcal{F})}$ such that for any $t \in \mathcal{T}_i$, $d(t) = i$. Then, the algorithm builds a decision tree processing layer one by one. Let $\mathcal{T}_i$ be such layer. We concatenate the (possibly disjoint) edges in T_i into a path P_i. Then, we compute an optimal edge ranking coloring of P_i using $\log n$ colors starting from color $(i-1) \cdot \log n$. Finally, we build a decision tree for the tests in $\mathcal{T}_i$ according to the edge ranking coloring. Observe that the resulting coloring is a valid edge ranking of P and that each edge has a color greater than all its predecessors in $\mathcal{F}$. Thus, the precedence constraints are respected. The final decision tree D is built recursively picking the root of the decision tree to be edge in P with the smallest color. it is easy to see that $\texttt{COST}_W(D, P) \leq h(\mathcal{F}) \cdot \log n$. Since the optimal decision tree has depth at least $h(\mathcal{F})$, the algorithm is an $O(\log n)$-approximation.

To obtain an algorithm for the average case, we use a different approach. A precedence constrained $1/2$-cut in a path is a subset of edges whose removal splits the path into subpaths of size at most $n/2$, such that the subset is precedence-closed. We wish to minimize the size of this set. We have the following observation: **troche nie widze tego lematu: co jesli precedensy to dwa lancuchy biegnace od srodka sciezki do jej konca/poczatku? Być może źle czytam, gdyz nie bylo zdefiniowane jak rozumiemy zbior krawedzi bedacy precedence closed?**

Lemma 5 *Let S^* be the optimal precedence constrained $1/2$-cut in a path $P = \langle v_1, \ldots, v_n \rangle$. Then either $S^* = F[\{(v_{\lfloor n/2 \rfloor} v_{\lfloor n/2 \rfloor + 1})\}]$ or there exist edges e_1, e_2 such that $S^* = F[\{e_1, e_2\}]$.*

Proof Observe that if S^* contains $(v_{\lfloor n/2 \rfloor} v_{\lfloor n/2 \rfloor + 1})$ then it is a $1/2$ cut. In such case we are done, since minimal precedence closed subset of edges containing $(v_{\lfloor n/2 \rfloor} v_{\lfloor n/2 \rfloor + 1})$ is $F[\{(v_{\lfloor n/2 \rfloor} v_{\lfloor n/2 \rfloor + 1})\}]$. Otherwise, let e_1 be the edge in S^* closest to v_1 and e_2 be the edge in S^* closest to v_n. Since e_1 and e_2 suffice to be a $1/2$-cut, the minimal precedence closed subset of edges with this property is $F[\{e_1, e_2\}]$. ■

Therefore one can easily compute such a cut in polynomial time by checking all possible candidates. We also have the following lemma relating the size of the optimal cut with the cost of the optimal decision tree:

Lemma 6 *Let P be the path induced by the instance $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ of the binary search problem with precedence constraints. Additionally, let $S^* \subseteq \mathcal{T}$ be the optimal precedence constrained $1/2$-cut in P. Then we have that $n \cdot |S^*| / 2 \leq \texttt{OPT}_A(I)$.*

Proof Let D^* be the optimal decision tree for I. Let S_D be the set of all tests which appeared at least once in D^* before the number of hypotheses consistent with previous responses dropped below $n/2$. Clearly, S_D is a precedence-closed $1/2$-cut in P. Additionally, each test in S_D contributes at least $n/2$ to the average cost of D^*. Thus, we have that $\texttt{COST}_A(D^*) \geq n \cdot |S_D| /2 \geq n \cdot |S^*| /2$. ■

The two lemmas above suggest a natural approach to finding a good strategy. Find the optimal precedence constrained $1/2$-cut S^* in P and use the tests in S^* to build any precedence-respecting decision tree D that splits the hypotheses into subproblems of size at most $n/2$. Then, recursively build decision trees for each subproblem and attach them to the leaves of D. It is easy to see that the resulting decision tree respects precedence constraints. We have the following theorem:

Theorem 7 *The above algorithm is an $O(\log n)$-approximation for the* PCACAL *in the case of binary search with precedence constraints.*

Proof Let D be the decision tree built by the algorithm. We prove by induction on n that $\frac{\texttt{COST}_A(D)}{\texttt{OPT}_A(I)} \leq 2\log n$. The base case for $n = 1$ is trivial since there is only one hypothesis and no tests are needed. For the inductive step, observe that the cost of D can be upper bounded as follows:

$$\begin{aligned}\frac{\texttt{COST}_A(D,P)}{\texttt{OPT}_A(I)} &\leq \frac{|S^*| \cdot n + \sum_{P' \in P - S^*} \texttt{COST}_A(D_{P'}, P')}{\texttt{OPT}_A(I)} \\ &\leq 2 + \frac{\sum_{P' \in P - S^*} 2 \cdot \log\left(\frac{n}{2}\right) \cdot \texttt{OPT}(I_{P'})}{\sum_{P' \in P - S^*} \texttt{OPT}(I_{P'})} \\ &= 2 + 2\log\left(\frac{n}{2}\right) = 2\log n.\end{aligned}$$

where the first inequality follows from the definition of average cost, the second inequality follows from the Lemma 6, inductive hypothesis, the fact that each subproblem has size at most $n/2$ and Lemma 2. The claim follows. ■

4. Active Learning via Covering Problems

To build our algorithm we will make use of the following definition:

Definition 8 (Sepcover) *Let D be any decision tree for $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. We define a sequence of tests P_D called* sepcover *as follows. Initially, P_D is empty and $\mathcal{H}' = \mathcal{H}$. While $|\mathcal{H}'| > |\mathcal{H}|/2$, we append to P_D the test $r(D_{\mathcal{H}'})$ and update $\mathcal{H}'$ to be the set of hypotheses corresponding to the child of $D_{\mathcal{H}'}$ that contains the most hypotheses. If $\texttt{COST}_W(D) = \texttt{OPT}_W(I)$, then we denote $P^*(I) = P_D$ (ties broken arbitrarily).*

It should be remarked that P_D is well-defined, as each test in P_D can have at most one child associated with more than half of the hypotheses in $\mathcal{H}'$. Since P_D is a subpath of D, we also have the following simple observation. Figure 4 illustrates the definition of sepcover.

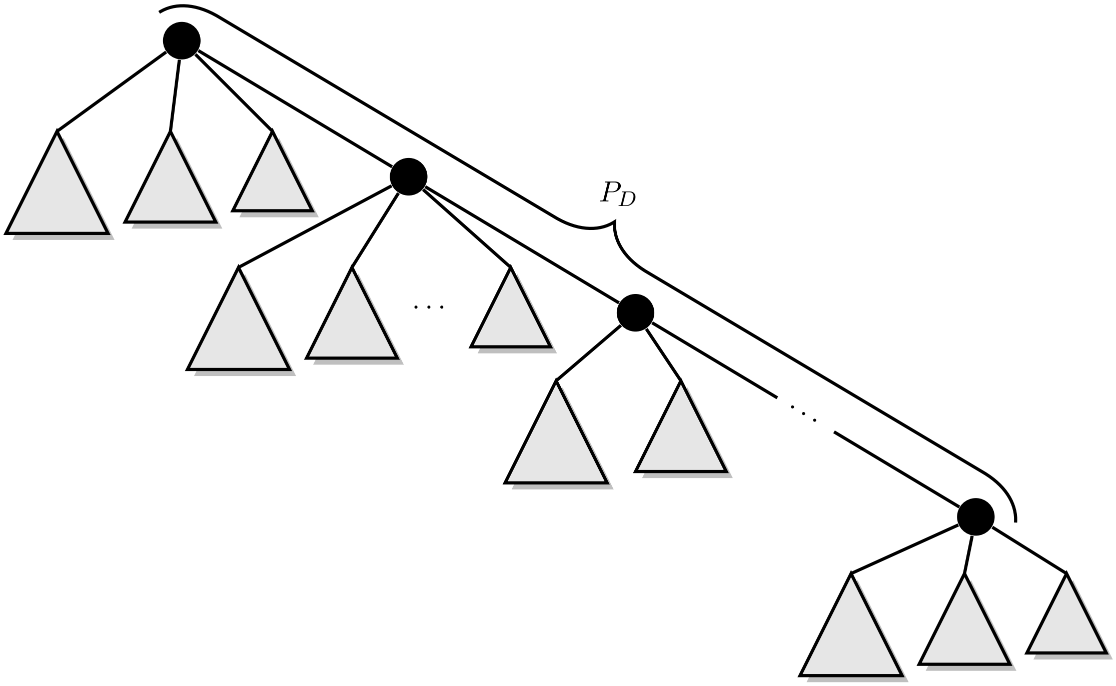


Figure 4: Sepcover sequence in a decision tree

We will say that a test in $P^*(I)$ *sepcovers* an element $u \in \mathcal{H}$ in C if after applying the test t in C, u belongs to a response $\mathcal{H}'$ of size at most $|\mathcal{H}|/2$. **tutaj jest cos dla mnie niejasne: nie wiem czy powyzsze definiuje P^*, gdyz jest to uzywane w lemacie ponizej i lemat nie mowi czy zachodzi dla kazdego P^* czy sa jakies warunki? W powyzszym takze z frazy “applyting the test t in C – to by sugerowalo, ze C zawiera jeden test (bo ”the“). Innymi slowy, nie wiem czym jest P^*.**

4.1. Worst Case

Observation 2 *Let I be any instance of . Then $|P^*(I)| \leq \texttt{OPT}_W(I)$.*

We will use $|P^*(I)|$ as a lower bound on $\texttt{OPT}_W(I)$ in the analysis of the approximation algorithm for . We have the following lemma:

Lemma 9 *Let $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ be any instance. Let S^* be the optimal solution for the PCSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ where $f = 1/4$ and a test t covers $h \in \mathcal{H}$ if $\{U_t(u)\} \leq \frac{3}{4} \cdot \{\mathcal{U}\}$. Then, $|S^*| \leq |P^*(I)|$.*

Proof We show that $P^*(I)$ is a feasible solution for the PCSC instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, K)$. Assume towards a contradiction that this is not the case, i. e. less than $|\mathcal{H}|/4$ are covered by $P^*(I)$. Therefore there exists $t \in P^*(I)$ and a response $\mathcal{H}'$ of size $|\mathcal{H}'| \leq |\mathcal{H}|/2$ such that hypotheses in $\mathcal{H}'$ are not covered by $P^*(I)$, otherwise the claim holds trivially, since all hypotheses are covered. Let $\mathcal{H}' \subseteq U_{t,j}$ (since $\mathcal{H}'$ is a response to a test t, such $U_{t,j}$ always exists). By assumption, we have that $|U_{t,j} - \mathcal{H}'| < |\mathcal{H}|/4$. Therefore, we have that $|U_{t,j}| = |\mathcal{H}'| + |U_{t,j} - \mathcal{H}'| < 3/4 \cdot |\mathcal{H}|$ which by definition means that h is covered by $P^*(I)$, a contradiction. ■

Theorem 10 *If there is an (γ, α)-bicriteria approximation algorithm for PCSC then there is an $O\left(\frac{\alpha}{\log\left(\frac{2\gamma}{2\gamma-1}\right)} \cdot \log n\right)$-approximation algorithm for . In particular when $\gamma = O(1)$, the approximation is $O(\alpha \cdot \log n)$.*

Proof The algorithm 1 is recursive and works as follows: Given an instance $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$, if $|\mathcal{H}| = 1$ we return the trivial decision tree with a single leaf corresponding to the only hypothesis in $\mathcal{H}$. Otherwise, we run the (γ, α)-approximation algorithm for PCSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$, where a test t covers element $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$. Let S be the returned set of tests. We build a decision tree D_S on tests from S closed under $\mathcal{F}$. For each $\mathcal{H}' \in \mathcal{H} - S$, we recursively call WORSTDECISIONTREE on instance $(\mathcal{H}', \mathcal{T} - S, \mathcal{F} - S)$ and attach the returned decision tree to the leaf of D_S corresponding to $\mathcal{H}'$. Finally, we return the constructed decision tree D. The following observation follows by Lemmas 1 and 9.:

```
Algorithm 1: The $O(\alpha \cdot \log n)$-approximation algorithm for the PCWCAL
procedure WORSTDECISIONTREE($\mathcal{H}, \mathcal{T}, \mathcal{F}$)
    if $|\mathcal{H}| = 1$ then
        return the trivial decision tree with a single leaf corresponding to the only hypothesis in $\mathcal{H}$.
    end
    foreach $t \in \mathcal{T}$ do
        Set $t$ to cover $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$.
    end
    $S \leftarrow$ Run the $(\gamma, \alpha)$-approximation algorithm for PCSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$.
    $D \leftarrow D_S \leftarrow$ any decision tree built on tests from $S$ respecting the precedence constraints $\mathcal{F}$.
    foreach $\mathcal{H}' \in \mathcal{H} - S$ do
        $D' \leftarrow$ WORSTDECISIONTREE($\mathcal{H}', \mathcal{T} - S, \mathcal{F} - S$).
        Attach $D'$ to the leaf of $D$ corresponding to $\mathcal{H}'$.
    end
    return $D$.
```

Observation 3 *Let D_S be the decision tree built on tests from S respecting the precedence constraints $\mathcal{F}$. Then, $\text{COST}_W(D_S) \leq \alpha \cdot |P^*(I)|$.*

We are now ready to prove the theorem.

Lemma 11 *Let D be the decision tree returned by* WORSTDECISIONTREE *on input $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. Then, $\mathtt{COST}_W(D, I) \leq \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n \cdot \mathtt{OPT}_W(I)$.*

Proof We prove the lemma by induction on n. The base case when $n = 1$ is trivial since the the cost of the decision tree is 0. Assume by induction that for every $I' = (\mathcal{H}', \mathcal{T}, \mathcal{F})$ such that $\mathcal{H}' \in \mathcal{H} - S$ and $n' = |\mathcal{H}'|$ we have $\mathtt{COST}_W(D', I') \leq \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n' \cdot \mathtt{OPT}_W(I')$, where D' is the decision tree returned by WORSTDECISIONTREE on input I'. We have that:

$$\begin{aligned}
\mathtt{COST}_W(D, I) \leq& \mathtt{COST}_W(D_S, I) + \max_{\mathcal{H}' \in \mathcal{H} - S} \mathtt{COST}_W\left(D', I'\right)\\
\leq& \alpha \cdot |S^*| + \max_{\mathcal{H}' \in \mathcal{H} - S} \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n' \cdot \mathtt{OPT}_W\left(I'\right)\\
\leq& \alpha \cdot |P^*(I)| + \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log\left(\frac{(4\gamma - 1) \cdot n}{4\gamma}\right) \cdot \mathtt{OPT}_W(I)\\
=& \alpha \cdot \mathtt{OPT}_W(I) + \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n \cdot \mathtt{OPT}_W(I) - \alpha \cdot \mathtt{OPT}_W(I)\\
=& \frac{\alpha}{\log\left(\frac{4\gamma}{4\gamma-1}\right)} \cdot \log n \cdot \mathtt{OPT}_W(I)
\end{aligned}$$

where the second inequality follows by the induction hypothesis, the third inequality follows by Lemma 1 and the fact that $|\mathcal{H}'| \leq \frac{(4\gamma-1)}{4\gamma} \cdot n$ for every $\mathcal{H}' \in \mathcal{H} - S$ and the equalities follow by rearranging terms. This concludes the proof of the lemma. ■

■

4.2. Average Case

We follow a similar idea, however we use the connection to PCMSSC instead of PCSC. In order to lower bound the cost of the optimal decision tree, we will need the following notion: Let S be any sequence of tests in a decision tree D. Then let:

$$\mathtt{COST}_A(S, I) = \sum_{t \in S} |\mathcal{H}_t| .$$

w powyższym nie widać jakie znaczenie ma wybór D.

We have the following observations:

Observation 4 *Let D be any decision tree for $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ and let S be any sequence of tests in D. Then,*

$$\mathtt{COST}_A(D, I) = \mathtt{COST}_A(S, I) + \sum_{D' \in D - S} \mathtt{COST}_A\left(D', I\right) .$$

As an immediate corollary for $D = D^*(I)$ and $S = P^*(I)$ we have:

Observation 5 *Let I be any instance of PCACAL. Then* $\texttt{COST}_A(P^*(I), I) \leq \texttt{OPT}_A(I)$.

This allows to use $\texttt{COST}_A(P^*(I), I)$ as a lower bound on $\texttt{OPT}_A(I)$ in the analysis of the approximation algorithm for PCACAL. We have the following lemma, analogous to Lemma 9.

Lemma 12 *Let $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$ be any PCACAL instance. Let S^* be the optimal solution for the PCMSSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$, where a test t covers element $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$. Then,* $\texttt{COST}_A(S^*) \leq \texttt{COST}_A(P^*(I))$.

Proof Assume towards a contradiction that $\texttt{COST}_A(S^*) > \texttt{COST}_A(P^*(I))$. We will show that in such case there exists a cover $\sigma \subseteq P^*(I)$ such that $c_A(\sigma) < \texttt{COST}_A(P^*(I))$, a contradiction with the optimality of S^*. Let σ be the smallest subsequence of $P^*(I)$ which covers at least $|\mathcal{H}|/4$ elements. Such a subsequence exists, by repeating the argument used in the proof of Lemma 9. Let $h \in \mathcal{H}$. There are two cases to consider:

- σ covers h. Consider the first test t that sepcovered h in $P^*(I)$. By definition, tests previous to t in σ cover at most $|\mathcal{H}|/4$ elements. Since at the moment of sepcovering, h belonged to a response of size at most $|\mathcal{H}|/2$, we know that t also covers h in σ. This means that the contribution of h to $c_A(\sigma)$ is at most its contribution to $\texttt{COST}_A(P^*(I))$.
- σ does not cover h. In such case h is sepcovered by some test t in $P^*(I)$ but not in σ. therefore, the contribution of h to $c_A(\sigma)$ is $|\sigma|$ and its contribution to $\texttt{COST}_A(P^*(I))$ is at least $|\sigma|$.

Thus, we have that $c_A(\sigma) < \texttt{COST}_A(P^*(I))$, a contradiction. ■

Theorem 13 *If there is a β- approximation algorithm for PCMSSC then there is an $O(\beta \cdot \log n)$-approximation algorithm for PCACAL.*

Proof The idea behind Algorithm 2 is the same as for the worst case version of the problem except the fact that we use a solution to PCMSSC instead of PCSC.

Algorithm 2: The $O(\beta \cdot \log n)$-approximation algorithm for the PCACAL

```
procedure AverageDecisionTree($\mathcal{H}, \mathcal{T}, \mathcal{F}$)
    if $|\mathcal{H}| = 1$ then
        return the trivial decision tree with a single leaf corresponding to the only hypothesis in $\mathcal{H}$.
    end
    foreach $t \in \mathcal{T}$ do
        Set $t$ to cover $u \in \mathcal{H}$ if for $u \in U_{t,j}$, $|U_{t,j}| \leq \frac{3}{4} \cdot |\mathcal{H}|$.
    end
    $S \leftarrow$ Run the $\beta$-approximation algorithm for PCMSSC on instance $(\mathcal{H}, \mathcal{T}, \mathcal{F}, f)$ with $f = 1/4$.
    $D \leftarrow D_S \leftarrow$ decision tree which consists of sequence of tests $S$.
    foreach $\mathcal{H}' \in \mathcal{H} - S$ do
        $D' \leftarrow$ AverageDecisionTree($\mathcal{H}', \mathcal{T} - S, \mathcal{F} - S$).
        Attach $D'$ to the leaf of $D$ corresponding to $\mathcal{H}'$.
    end
    return $D$.
```

Observation 6 *Let D_S be the decision tree consisting of the sequence of tests S. Then, $\mathit{COST}_A(D_S, I) \leq \beta \cdot \mathit{COST}_A(P^*(I)\,, I)$.*

We are now ready to prove the theorem:

Lemma 14 *Let D be the decision tree returned by* AVERAGEDECISIONTREE *on input $I = (\mathcal{H}, \mathcal{T}, \mathcal{F})$. Then, $\mathit{COST}_A(D, I) \leq \beta \cdot \log_{4/3} n \cdot \mathit{OPT}_A(I)$.*

Proof We prove the lemma by induction on n. The base case when $n = 1$ is trivial since the the cost of the decision tree is 0. Assume by induction that for every $I' = (\mathcal{H}', \mathcal{T}, \mathcal{F})$ such that $\mathcal{H}' \in \mathcal{H} - S$ and $n' = |\mathcal{H}'|$ we have $\mathtt{COST}_A(D', I') \leq \beta \cdot \log_{4/3} n' \cdot \mathtt{OPT}_A(I')$, where D' is the decision tree returned by AVERAGEDECISIONTREE on input I'. We have that:

$$\begin{aligned}
\mathtt{COST}_A(D, I) \leq & \mathtt{COST}_A(D_S, I) + \sum_{\mathcal{H}' \in \mathcal{H} - S} \mathtt{COST}_A\big(D', I'\big) \\
\leq & \beta \cdot \mathtt{COST}_A(P^*(I)\,, I) + \sum_{\mathcal{H}' \in \mathcal{H} - S} \beta \cdot \log_{4/3} n' \cdot \mathtt{OPT}_A\big(I'\big) \\
\leq & \beta \cdot \mathtt{COST}_A(P^*(I)\,, I) + \sum_{\mathcal{H}' \in \mathcal{H} - S} \beta \cdot \log_{4/3}\left(\frac{3}{4} \cdot n\right) \cdot \mathtt{OPT}_A(I) \\
= & \beta \cdot \mathtt{COST}_A(P^*(I)\,, I) + \beta \cdot \Big(\log_{4/3} n - 1\Big) \cdot \mathtt{OPT}_A(I) \\
\leq & \beta \cdot \log_{4/3} n \cdot \mathtt{OPT}_A(I)
\end{aligned}$$

where the second inequality follows by the induction hypothesis, the third inequality follows by Lemma 1 and the fact that $|\mathcal{H}'| \leq \frac{3}{4} \cdot n$ for every $\mathcal{H}' \in \mathcal{H} - S$ and the equalities follow by rearranging terms. This concludes the proof of the lemma. ■

■

5. Set covering with constraints

5.1. Max-Density Precedence-Closed Subfamily (MDPCS)

The key to solce PCSC and PCMSSC is to solve the MDPCS problem. An approximation algorithm for MDPCS can be used as an essential subroutine in our algorithms for PCSC and PCMSSC. By **?**, the following greedy algorithm achieves an $O(\sqrt{m})$-approximation for MDPCS:

Algorithm 3: The greedy algorithm for MDPCS

```
procedure MDPCS-Greedy($\mathcal{G}, \prec, R$)
    $\mathcal{A} \leftarrow \mathcal{G}$
    foreach $S \in \mathcal{G}$ do
        if $\Delta(P[S], R) > \Delta(\mathcal{A}, R)$ then
            $\mathcal{A} \leftarrow P[S]$
        end
    end
    return $\mathcal{A}$
```

Let $\delta = \max_{S \in \mathcal{G}} \Delta(P[S], R)$. When $\delta \geq 1$, then the approximation factor of the greedy can also be bounded by $O(\sqrt{n})$. We show that if we enforce a certain condition on the input called *ϵ-shallow ancestry*, then for $\epsilon < 1$ the greedy algorithm achieves an $O(n^\epsilon)$-approximation .

For each $S \in \mathcal{G}$, let $p(S) = |P[S]|$ and $c(S, R) = |\mathrm{cov}(P[S], R)|$.

Theorem 15 *Suppose there exists a constant $C > 0$ and $\epsilon \in (0, 1)$ such that for all $S \in \mathcal{G}$, $p(S) \leq C \cdot c(S)^\epsilon$ (ϵ-shallow ancestry). Then MDPCS-Greedy provides an $O(n^\epsilon)$-approximation.*

Proof Let $\mathcal{A}^*$ be an optimal solution consisting of sets $S_1, \ldots, S_k$. There are two cases:

1. If $\delta \geq n^{1-\epsilon}$, then, we observe that $\Delta(\mathcal{A}, R) = \delta \geq n^{1-\epsilon}$. Since we can cover at most n elements with at least one set, we have $\Delta(\mathcal{A}^*, R) \leq n$. Therefore:

$$\frac{\Delta(\mathcal{A}^*, R)}{\Delta(\mathcal{A}, R)} \leq \frac{n}{n^{1-\epsilon}} = n^\epsilon$$

2. Else, if $\delta \leq n^{1-\epsilon}$, we proceed as follows: By definition of density, for any $S \in \mathcal{G}$, $c(S) \leq \delta \cdot p(S)$. Combining this with the ϵ-shallow ancestry condition, we have that for all $S \in \mathcal{G}$, $c(S) \leq \delta \cdot C \cdot c(S)^\epsilon$. Rearraning this inequality, we get that $c(S) \leq (\delta \cdot C)^{\frac{1}{1-\epsilon}}$. We have that:

$$|\mathrm{cov}(\mathcal{A}^*)| = \left| \bigcup_{j=1}^{k} \mathrm{cov}(S_j, R) \right| \leq \sum_{j=1}^{k} c(S_j) \leq k \cdot (\delta \cdot C)^{\frac{1}{1-\epsilon}}$$

Therefore:

$$\Delta(\mathcal{A}^*, R) = \frac{|\mathrm{cov}(\mathcal{A}^*, R)|}{k} \leq (\delta \cdot C)^{\frac{1}{1-\epsilon}}$$

By the greedy choice, $\Delta(\mathcal{A}, R) \geq \delta$ and by assumption $\delta \leq n^{1-\epsilon}$. Thus:

$$\frac{\Delta(\mathcal{A}^*, R)}{\Delta(\mathcal{A}, R)} \leq \frac{(\delta \cdot C)^{\frac{1}{1-\epsilon}}}{\delta} = C^{\frac{1}{1-\epsilon}} \cdot \delta^{\frac{\epsilon}{1-\epsilon}} \leq C^{\frac{1}{1-\epsilon}} \cdot (n^{1-\epsilon})^{\frac{\epsilon}{1-\epsilon}} = C^{\frac{1}{1-\epsilon}} \cdot n^\epsilon$$

Since C is constant, the theorem follows. ■

5.2. Precedence constrained set cover

Tu poniżej miałem jakąś próbe pisania tego, ale sie pokomplikowało więc ten pseuudokod jest niekompletny. To jest do zmiany wszystko. Zostawiam na razie te sekcje tobie Darku.

We show the following:

Theorem 16 *If there exists an γ approximation algorithm for the MDPCS problem, then there exists an $(H_K + 1) \cdot \gamma$ - approximate algorithm for the PCSC problem.*

Proof ■

Algorithm 4: The γ-greedy algorithm for PCSC

```
procedure PCSC($\mathcal{U}, \mathcal{S}, \mathcal{F}, K$)
    $\mathcal{C} \leftarrow \emptyset$
    while $|\mathrm{cov}(\mathcal{C},\mathcal{U})| < K$ do
        $\mathcal{A} \leftarrow$ Run the $\gamma$-approx. algorithm for MDPCS on $(\mathcal{U} - \mathcal{C}, \mathcal{S} - \mathcal{C}, \mathcal{F} - \mathcal{C}, m)$
        if $|\mathrm{cov}(\mathcal{C} \cup \mathcal{A},\mathcal{U})| \geq K$ then
            Find the minimum budget $B \in [|\mathcal{A}|]$, such that the $\gamma$-approx. algorithm for MDPCS on
            $(\mathcal{U} - \mathcal{C}, \mathcal{S} - \mathcal{C}, \mathcal{F} - \mathcal{C}, B)$ returns a set $\mathcal{B}$ with $|\mathrm{cov}(\mathcal{B},\mathcal{U} - \mathcal{C})| \geq \frac{K - \mathrm{cov}(\mathcal{C},\mathcal{U})}{\alpha}$
            while $|\mathrm{cov}(\mathcal{C},\mathcal{U})| < K$ do
                $\mathcal{B} \leftarrow$ Run the $\gamma$-approx. algorithm for MDPCS on $(\mathcal{U} - \mathcal{C}, \mathcal{S} - \mathcal{C}, \mathcal{F} - \mathcal{C}, B)$
                $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{B}$
            end
            return $\mathcal{C}$
        end
        foreach $u \in \mathrm{cov}(\mathcal{A},\mathcal{U} - \mathcal{C})$ do
            $c(u) \leftarrow \Delta(\mathcal{A},\mathcal{U} - \mathcal{C})$
        end
        $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{A}$
    end
```

Theorem 17 *If the precedence constraints form an outforest, then there exists an bicriteria $(4, O(\log n))$-approximation algorithm for PCSC which be converted to an $O\big(\log^2 n\big)$ approximation algorithm.*

5.3. Precedence constrained min sum set cover

By **?**:

Theorem 18 *If there exists an γ approximation algorithm for the MDPCS problem, then there exists an $4 \cdot \gamma$ - approximate algorithm for the PCMSSC problem.*

Theorem 19 *If the precedence constraints form an outforest, then there exists an $(O(\log n))$-approximation algorithm for PCMSSC.*

6. Hardness

6.1. Binary search with precedence constraints

We begin by showing that the active learning problem is NP-hard even for the very special of binary searching with precedence. In this problem we are given a linearly ordered set of hypotheses $h_1 \prec h_2 \prec \ldots \prec h_n$ and tests that correspond to binary queries of the form "Is the unknown hypothesis h_i less than or equal to h_k?". Additionally, a precedence constraints on queries are given which enforce certain queries to be performed before others.

Interestingly, this problem is connected to a seemingly unrelated problem of parallel evaluation of arithmetic expressions. In this problem we are given an arithmetic expression consisting of n variables combined using arbitrary binary operations. Additionally, due to the order of operations of the usage of brackets, a precedence constraints on procesing the operations is given. The goals is to evaluate this expression using arbitrary number of parallel processors however it is required that any operand can undergo only one operation at a time and the precedence constraints are satisfied. The goal is to minimize the total time required to evaluate the expression. It is widely known that without the precedence constraints this problem is equivalent to binary searching on a linearly ordered set of hypotheses. This connection is still valid even when precedence constraints are present, however the precedence constraints on the operations must be reversed. In what follows we show that the problem of binary searching with precedence is NP-hard thereby implying that parallel evaluation of arithmetic expressions with precedence constraints is NP-hard as well.

Theorem 20 *Worst-Case Binary Search With Precedence Constraints is NP-hard.*

Theorem 21 *Average-Case Binary Search With Precedence Constraints is NP-hard.*

6.2. Reducing set covering to active learning

To show inapproximability results for PCWCAL and PCACAL, we exploit a well known reduction from Set Cover to the problem of Active Learning. Although similar reductions are known in the literature for the unprecedented case Chakaravarthy et al. (2007); Laber and Nogueira (2004); Cicalese et al. (2014) we show that it can also be established even when precedence constraints are present. Therefore to obtain inapproximability results for PCWCAL and PCACAL, it suffices to show inapproximability results for the relevant version of Set Cover with precedence constraints.

Theorem 22 *If there is no α-approximation for PCSC, then there is no $O(\alpha)$-approximation for PCWCAL.*

Proof Let $(U, \mathcal{S}, P)$ be an instance of PCSC where U is the universe of elements, $\mathcal{S}$ is the family of sets and P are the precedence constraints on $\mathcal{S}$. We construct an instance of PCWCAL as follows. For each element $u \in U$, we create two representative hypotheses h_u^0 and h_u^1. For each set $S \in \mathcal{S}$, we create a test t_S such that $t_S(h_u^i) = h_u^i$ if $u \in S$ and $t_S(h_u^i) = 0$ otherwise. The precedence constraints on tests are the same as the precedence constraints on sets, i.e., if S_1 must be selected before S_2 in PCSC, then t_{S_1} must be selected before t_{S_2} in PCWCAL. Notice, that any valid decision tree for the constructed instance of PCWCAL is a path of tests (ignoring leafs) since performing any test t_S either returns a response containing one of the hypotheses h_u^i, thus ending a process or returns 0, which is a singular possible response leading to the next test. Note that even when the candidate set contains only representatives of one element u, we still need to perform a test to distinguish between h_u^0 and h_u^1, which could not be a case if there was only one hypothesis per element. Therefore, any valid decision tree for the constructed instance of PCWCAL corresponds to a valid selection of sets in PCSC and vice versa. Moreover, the cost of the decision tree is equal to the number of selected sets. Thus, since the size of the reduction is linear any α-approximation for PCWCAL would yield an $O(\alpha)$-approximation for PCSC. ■

Theorem 23 *If there is no α-approximation for PCMSSC, then there is no $O(\alpha)$-approximation for PCACAL.*

Proof We use the same reduction as in the previous theorem. Note that since we doubled the number of hypotheses in the construction, the cost of any decision tree in the constructed instance of PCACAL is twice the number of selected sets in PCMSC. However, this does not affect the structure of the reduction and thus any α-approximation for PCACAL would yield an $O(\alpha)$-approximation for PCMSSC. ■

6.3. Outforest precedence constraints

Theorem 24 *PCWCAL with outforest precedence constraints is NP-hard to approximate within a factor of $O\big(\log^{2-\epsilon} n\big)$ for any $\epsilon > 0$ unless NP $\subseteq$ ZTIME$\big(n^{polylog(n)}\big)$.*

Proof To show this we prove that Group Steiner Tree on tree metrics is reducible to Set Cover with outforest precedence constraints. Since GST on trees cannot be approximated within a factor of $O\big(\log^2 n\big)$ unless NP $\subseteq$ ZTIME$\big(n^{\text{polylog}(n)}\big)$ Halperin and Krauthgamer (2003). Note that in the latter reduction all of the weights are of form 2^{-h} with exponent ranging from 0 to $h = O\big(\log^{1-\epsilon} n\big)$. Therefore, we can scale all weights by $2^h = \text{poly}(n)$ to obtain integer weights without affecting the approximation ratio. Let $T, w, \mathcal{G}$ be an instance of GST where T is a tree metric with root r, w are the weights on the edges of T and $\mathcal{G} = (G_1, \ldots, G_k)$ are the groups. We construct an instance of PCSC with outforest precedence constraints as follows. Since to include a node in the Steiner Tree we need to include its parent edge of cost w_e, for each vertex $v \neq r$ we create its w_e representatives $S_v^1 \preceq \ldots \preceq S_v^{w_e}$ so that taking S^{w_e} to the cover requires taking all of the previous representatives. Additionally, for any directed edge $uv \in E$ such that $u \neq r$ and e is a parent edge of u, we set $S_u^{w_e} \preceq S_v^1$ to enforce the condition that to including a node in a Steiner Tree requires including its parent node with its parent edge. For each group G_i we create a universe element u_i. If $v \in G_i$ and e is the parent edge of v, we set S_{w_e} to cover u_i. It is easy to see that in order to cover any universe element by a vertex v one needs to include all representatives of ancestors of v excluding r. Therefore, any valid selection of sets in the constructed instance of PCSC corresponds to a valid Steiner Tree in the instance of GST and vice versa. Moreover, the cost of the selected sets is equal to the cost of the Steiner Tree. Thus, PCSC with outforest precedence constraints cannot be approximated within a factor of $O\big(\log^{2-\epsilon} n\big)$ for any $\epsilon > 0$ unless NP $\subseteq$ ZTIME$\big(n^{\text{polylog}(n)}\big)$ and the same holds for PCWCAL with outforest precedence constraints. ■

6.4. General precedence constraints

In this section we show strong inapproximability results for PCWCAL and PCACAL with general precedence constraints by reducing from the Planted Dense Subgraph Conjecture which is a widely believed statement about hardness of detecting a dense component within an Erdős-Renyi graph Charikar et al. (2016); McClintock et al. (2017).

The Planted Dense Subgraph Conjecture states that for any constants $\beta < \alpha$ and any $k \geq \sqrt{N}$, there is no polynomial time algorithm that can distinguish between the following two distributions of graphs with any advantage $\epsilon > 0$:

- With probability 1/2, G_1: an Erdős-Renyi graph $G(N, N^{\alpha-1})$,
- With probability 1/2, G_2: an Erdős-Renyi graph $G(N, N^{\alpha-1})$ with a planted subgraph of size k and edge density $k^{\beta-1}$.

Using this conjecture one can show the following inapproximability results for PCMSSC McClintock et al. (2017):

Theorem 25 *For any $\epsilon > 0$ PCMSSC cannot be approximated within a factor of $O\big(m^{1/6-\epsilon}\big)$ nor $o\big(n^{1/12-\epsilon}\big)$ condition to Planted Dense Subgraph Conjecture.*

Using the reduction from PCMSSC to PCACAL and PCWCAL from the previous section we immediately obtain the following results:

Theorem 26 *For any $\epsilon > 0$ PCACAL cannot be approximated within a factor of $O\big(m^{1/6-\epsilon}\big)$ nor $o\big(n^{1/12-\epsilon}\big)$ condition to Planted Dense Subgraph Conjecture.*

Below we show that a similar reduction can also be used to obtain the same inapproximability result for PCSC. In the reduction we will often make use of the Chernoff Bound which is as follows:

Theorem 27 (Chernoff Bound) *Let $X_1, X_2, \ldots, X_n$ be independent random variables taking values in $\{0, 1\}$ such that for every $i \in [n]$, $\mathbb{P}[X_i = 1] = p$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = \mathbb{E}[X] = pn$. Then, for every $\delta \in (0, 1)$ it holds that:*

$$\mathbb{P}[X \leq (1-\delta)\mu] \leq e^{-\frac{\delta^2\mu}{2}},$$
$$\mathbb{P}[X \geq (1+\delta)\mu] \leq e^{-\frac{\delta^2\mu}{3}}.$$

We have the following theorem:

Theorem 28 *For any $\epsilon > 0$ PCSC cannot be approximated within a factor of $O\big(m^{1/6-\epsilon}\big)$ nor $o\big(n^{1/12-\epsilon}\big)$ condition to Planted Dense Subgraph Conjecture.*

Proof We start by choosing appropriate parameters for our reduction. Let $k = \sqrt{N}$, $\alpha = 1/2$ and $\beta = 1/2 - \gamma$. In our reduction we will embded the structure of the graph into the precedence constraints, while the universe elements will merely enforce the covering requirement. Given a graph $G = \{[N], \mathcal{E}\}$ we firstly convert each vertex v of G into λ representative sets $V_{u,i}$ for $i \in [\lambda]$ where λ is appropriately chosen natural number. We then convert each edge $uv \in \mathcal{E}$ into one representative set E_{uv} which is a successor of all representatives of u and v, i.e., $V_{u,i} \preceq E_{uv}$ and $V_{v,i} \preceq E_{uv}$ for all $i \in [\lambda]$.

We then create the universe elements $U = \{0, \ldots, n\}$ for some appropriately chosen natural number n. Every vertex representative $V_{u,i}$ covers consists of a singular item $V_{u,i} = \{0\}$. In order to define the edge representatives we would firstly like to associate items with vertices. To do so we built auxiliary sets U_v for each vertex v. The construction is randomized, and based on a probability $p \in (0,1)$. Each item in $[n]$ is associated with U_v independently with probability p, so that $\mathbb{P}[j \in U_v] = p$ for each $j \in [n]$. Finally, each edge representative E_{uv} covers the union of the auxiliary sets of its endpoints, i.e., $E_{uv} = U_u \cup U_v$. We have that $\mathbb{E}[|U_v|] = pn$, $\mathbb{E}[|v\colon j \in U_v|] = pN$ and $\mathbb{E}[|E_{uv}|] = \mathbb{E}[|U_u \cup U_v|] = np^2$.

The idea is as follows: We pick p to be smallest value for which if planted component exists then with high probability it covers U in which case we can showcase a good solution to PCSC. On the other hand, if no planted component exists then we show that it pushes the cost of any solution to PCSC high enough to obtain the desired inapproximability ratio.

Let $\mathcal{P}$ be the planted component if it exists. Since $\mathcal{P}$ consists of $\sqrt{N}$ vertices we have that $\mathbb{E}[|v\colon j \in U_v, v \in V\{\mathcal{P}\}|] = p\sqrt{N}$. Therefore, we also have that

$$\mathbb{E}[|\{v,u\} : j \in U_v, j \in U_u, v, u \in V\{\mathcal{P}\}|] = \binom{p\sqrt{N}}{2}.$$

Since each edge in $\mathcal{P}$ exists independently with probability $k^{\beta-1} = \sqrt{N}^{-1/2-\gamma} = N^{-1/4-\gamma/2}$ we have that $\mathbb{E}[|E_{uv}\colon j \in E_{uv}|] = \binom{p\sqrt{N}}{2} \cdot N^{-1/4-\gamma/2} \geq p^2N/4 \cdot N^{-1/4-\gamma/2} = N^{3/4-\gamma/2}p^2$. So that this is large enough we pick $p = 32 \cdot N^{-3/8+\gamma/4} \cdot \log n$.

Assume that the planted component exists. Then we show that by taking all edge representatives corresponding to edges in the planted component (and all of their predecessors) we can cheapily cover all items in U with high probability. Since we have that $\mathbb{E}[|v\colon j \in U_v, v \in V\{\mathcal{P}\}|] = p\sqrt{N}$ by Chernoff's bound we have that:

$$\mathbb{P}[|v\colon j \in U_v, v \in V\{\mathcal{P}\}| \leq p\sqrt{N}/2] \leq e^{-p\sqrt{N}/8} = e^{-4\cdot N^{1/8+\gamma/4}\cdot \log n} = n^{-4\cdot N^{1/8+\gamma/4}}.$$

which is extremely small. Since for large enough x, $\binom{x}{2} \geq x^2/4$ we have that:

$$\mathbb{P}[|uv\colon j \in U_v, j \in U_u, v, u \in V\{\mathcal{P}\}| \leq p^2N/16] \leq e^{-4\cdot N^{1/8+\gamma/4}\cdot \log n} = n^{-4\cdot N^{1/8+\gamma/4}}.$$

Since each edge in $\mathcal{P}$ exists independently with probability $N^{-1/4-\gamma/2}$ we have that:

$$\mathbb{E}[|E_{uv}\colon j \in E_{uv}, uv \in V(\mathcal{P})|] \geq p^2N/16 \cdot N^{-1/4-\gamma/2} = 4 \cdot N^{3/4-\gamma/2}p^2 = 64\log^2 N.$$

Let $\mu = \mathbb{E}[|E_{uv}\colon j \in E_{uv}, uv \in V(\mathcal{P})|]$. We have that:

$$\mathbb{P}[|E_{uv}\colon j \in E_{uv}, uv \in V(\mathcal{P})| \leq \mu/2] \leq e^{-\mu/8} = e^{-8\log^2 N} = N^{-8\log N}.$$

By applying the union bound, the probability that there exists an item $j \in [n]$ that is not covered by the representatives of the planted component is at most $n/N^{8\log N}$ which by our (future) choice of $n = O(N)$ is very small. Therefore, with high probability all items in U are covered by taking all edge representatives corresponding to edges in the planted component (and all of their predecessors). Observe that:

$$\mathbb{E}[|E(\mathcal{P})|] = \binom{\sqrt{N}}{2} \cdot N^{-1/4-\gamma/2} \leq N^{3/4-\gamma/2}/2.$$

Again, using Chernoff's bound we have that:

$$\mathbb{P}[|E(\mathcal{P})| \geq N^{3/4-\gamma/2}] \leq e^{-N^{3/4-\gamma/2}/6}.$$

Therefore, with high probability the cost of the solution is at most $\lambda\sqrt{N} + N^{3/4-\gamma/2}$. Now assume that the planted component does not exist. We show that any solution to PCSC must have high cost with high probability.

Tutaj dowod jest jeszcze enigmatyczny dla mnie dlatego zostawiam to na razie. TODO: pokazac, ze jak nie ma komponentu to koszt rozwiazania mocno rosnie ■

Therefore we immediately have that:

Theorem 29 *For any $\epsilon > 0$ PCWCAL cannot be approximated within a factor of $O\big(m^{1/6-\epsilon}\big)$ nor $o\big(n^{1/12-\epsilon}\big)$ condition to Planted Dense Subgraph Conjecture.*

7. Conclusions and Future Work

References

Venkatesan T. Chakaravarthy, Vinayaka Pandit, Sambuddha Roy, Pranjal Awasthi, and Mukesh Mohania. Decision trees for entity identification: approximation algorithms and hardness results. In *Proceedings of the Twenty-Sixth ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '07, page 53–62, New York, NY, USA, 2007. Association for Computing Machinery. ISBN 9781595936851. doi: 10.1145/1265530.1265538. URL https://doi.org/10.1145/1265530.1265538.

Moses Charikar, Yonatan Naamad, and Anthony Wirth. On Approximating Target Set Selection. In Klaus Jansen, Claire Mathieu, José D. P. Rolim, and Chris Umans, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2016)*, volume 60 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 4:1–4:16, Dagstuhl, Germany, 2016. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-018-7. doi: 10.4230/LIPIcs.APPROX-RANDOM.2016.4. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX-RANDOM.2016.4.

Ferdinando Cicalese, Eduardo Laber, and Aline Medeiros Saettler. Diagnosis determination: decision trees optimizing simultaneously worst and expected testing cost. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, page I–414–I–422. JMLR.org, 2014.

Eran Halperin and Robert Krauthgamer. Polylogarithmic inapproximability. In *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, STOC '03, page 585–594, New York, NY, USA, 2003. Association for Computing Machinery. ISBN 1581136749. doi: 10.1145/780542.780628. URL https://doi.org/10.1145/780542.780628.

Eduardo S. Laber and Loana Tito Nogueira. On the hardness of the minimum height decision tree problem. *Discrete Applied Mathematics*, 144(1):209–212, 2004. ISSN 0166-218X. doi: https://doi.org/10.1016/j.dam.2004.06.002. URL https://www.sciencedirect.com/science/article/pii/S0166218X04001787. Discrete Mathematics and Data Mining.

Tak Wah Lam and Fung Ling Yue. Optimal edge ranking of trees in linear time. *Algorithmica*, 30 (1):12–33, 2001. doi: 10.1007/s004530010076.

Jessica McClintock, Julián Mestre, and Anthony Wirth. Precedence-Constrained Min Sum Set Cover. In Yoshio Okamoto and Takeshi Tokuyama, editors, *28th International Symposium on Algorithms and Computation (ISAAC 2017)*, volume 92 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 55:1–55:12, Dagstuhl, Germany, 2017. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. ISBN 978-3-95977-054-5. doi: 10.4230/LIPIcs.ISAAC.2017.55. URL https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ISAAC.2017.55.

Appendix A. My Proof of Theorem 1

This is a boring technical proof.

Appendix B. My Proof of Theorem 2

This is a complete version of a proof sketched in the main text.