Approximating Red-Blue Set Cover and Minimum Monotone Satisfying Assignment

Eden Chlamtáč
Department of Computer Science, Ben-Gurion University of the Negev, Beer-Sheva, Israel

Yury Makarychev
Toyota Technological Institute at Chicago (TTIC), IL, USA

Ali Vakilian
Toyota Technological Institute at Chicago (TTIC), IL, USA

Abstract

We provide new approximation algorithms for the Red-Blue Set Cover and Circuit Minimum Monotone Satisfying Assignment (MMSA) problems. Our algorithm for Red-Blue Set Cover achieves $\tilde{O}(m^{1/3})$-approximation improving on the $\tilde{O}(m^{1/2})$-approximation due to Elkin and Peleg (where m is the number of sets). Our approximation algorithm for MMSA$_t$ (for circuits of depth t) gives an $\tilde{O}(N^{1-\delta})$ approximation for $\delta = \frac{1}{3}2^{3-\lceil t/2 \rceil}$, where N is the number of gates and variables. No non-trivial approximation algorithms for MMSA$_t$ with $t \geq 4$ were previously known.

We complement these results with lower bounds for these problems: For Red-Blue Set Cover, we provide a nearly approximation preserving reduction from Min k-Union that gives an $\tilde{\Omega}(m^{1/4-\varepsilon})$ hardness under the Dense-vs-Random conjecture, while for MMSA we sketch a proof that an SDP relaxation strengthened by Sherali–Adams has an integrality gap of $N^{1-\varepsilon}$ where $\varepsilon \to 0$ as the circuit depth $t \to \infty$.

2012 ACM Subject Classification Theory of computation → Approximation algorithms analysis; Theory of computation → Circuit complexity

Keywords and phrases Red-Blue Set Cover Problem, Circuit Minimum Monotone Satisfying Assignment (MMSA) Problem, LP Rounding

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2023.11

Category APPROX

Related Version *Full Version*: `https://arxiv.org/abs/2302.00213`

Funding *Eden Chlamtáč*: The work was done while the author was visiting and supported by TTIC.
Yury Makarychev: Supported by NSF awards CCF-1955173 and CCF-1934843.

1 Introduction

In this paper, we study two problems, *Red-Blue Set Cover* and its generalization *Circuit Minimum Monotone Satisfying Assignment*. Red-Blue Set Cover, a natural generalization of Set Cover, was introduced by Carr et al. [5]. Circuit Minimum Monotone Satisfying Assignment, a problem more closely related to Label Cover, was introduced by Alekhnovich et al. [2] and Goldwasser and Motwani [12].

▶ **Definition 1.** *In Red-Blue Set Cover, we are given a universe of $(k+n)$ elements U partitioned into disjoint sets of red elements (R) of size n and blue elements (B) of size k, that is $U = R \cup B$ and $R \cap B = \emptyset$, and a collection of sets $\mathcal{S} := \{S_1, \cdots, S_m\}$. The goal is to find a sub-collection of sets $\mathcal{F} \subseteq \mathcal{S}$ such that the union of the sets in $\mathcal{F}$ covers all blue elements while minimizing the number of covered red elements.*

Besides Red-Blue Set Cover, we consider the Partial Red-Blue Set Cover *problem in which we are additionally given a parameter $\hat{k}$, and the goal is cover at least $\hat{k}$ blue elements while minimizing the number of covered red elements.*

▶ **Definition 2.** *The Circuit Minimum Monotone Satisfying Assignment problem of depth t, denoted as $MMSA_t$, is as follows. We are given a circuit C of depth t over Boolean variables $x_1, \ldots, x_n$. Circuit C has AND and OR gates: all gates at even distances from the root (including the output gate at the root) are AND gates; all gates at odd distances are OR gates. The goal is to find a satisfying assignment with the minimum number of variables x_i set to 1 (true).*

Note that C computes a monotone function and the assignment of all ones is always a feasible solution. Though the definitions of the problems are quite different, Red-Blue Set Cover and MMSA_t are closely related. Namely, Red-Blue Set Cover is equivalent to MMSA_3.[1] The correspondence is as follows: variables $x_1, \ldots, x_n$ represent red elements; AND gates in the third layer represent sets $S_1, \ldots, S_m$; OR gates in the second layer represent blue elements. The gate for a set S_j is connected to OR gates representing blue elements of S_j and variables x_i representing red elements of S_j. It is easy to see that an assignment to $x_1, \ldots, x_n$ satisfies the circuit if and only if there exists a sub-family $\mathcal{F} \subseteq \mathcal{S}$ that covers all the blue elements, and only covers red elements corresponding to variables x_i which are assigned 1 (but not necessarily all of them).

Background. Red-Blue Set Cover and its variants are related to several well-known problems in combinatorial optimization including *group Steiner* and *directed Steiner* problems, *minimum monotone satisfying assignment* and *ymmetric minimum label cover*. Arguably, the interest to the general MMSA_t problem is mostly motivated by its connection to complexity and hardness of approximation.

The Red-Blue Set Cover has applications in various settings such as *anomaly detection, information retrieval* and notably in *learning of disjunctions* [5]. Learning of disjunctions over $\{0,1\}^m$ is one of the basic problems in the PAC model. In this problem, given an arbitrary distribution $\mathcal{D}$ over $\{0,1\}^m$ and a target function $h^* : \{0,1\}^m \to \{-1,+1\}$ which denotes the true labels of examples, the goal is to find the best disjunction $f^* : \{0,1\}^m \to \{-1,+1\}$ with respect to $\mathcal{D}$ and h^* (i.e., $f^*(x)$ computes a disjunction of a subset of coordinates of x). The problem of learning disjunctions can be formulated as an instance of the (Partial) Red-Blue Set Cover problem [4]: we can think of the positive examples as blue elements (i.e., $h^*(x) = 1$) and the negative examples as red elements (i.e., $h^*(x) = -1$). Then, we construct a set S_i corresponding to each coordinate i and the set S_i contains an example x if the i-th coordinate of x is equal to 1. Let $C \subset \{S_1, \cdots, S_m\}$. Then, the disjunction f_C corresponding to C, i.e., $f_C(x) := \bigvee_{S_i \in C} x_i$, outputs one on an example x if in the constructed Red-Blue Set Cover instance, the element corresponding to x is covered by sets in C.

As we observe in Section 5, Red-Blue Set Cover is also related to the Min k-Union problem which is a generalization of Densest k-Subgraph [8]. In Min k-Union, given a collection of m sets $\mathcal{S}$ and a target number of sets k, the goal is to pick k sets from $\mathcal{S}$ whose union has the minimum cardinality. Notably, under a hardness assumption, which is an extension of the "Dense vs Random" conjecture for Densest k-Subgraph to hypergraphs, Min k-Union

[1] Also observe that MMSA_2 is equivalent to Set Cover.

cannot be approximated better than $\tilde{\Omega}(m^{1/4-\varepsilon})$ [9]. In this paper, we prove a hardness of approximation result for Red-Blue Set Cover by constructing a reduction from Min k-Union to Red-Blue Set Cover.

1.1 Related work

Carr et al. [5] formulated the Red-Blue Set Cover problem and presented a $2\sqrt{m}$-approximation algorithm for the problem when every set contains only one blue element. Later, Elkin and Peleg [11] showed that it is possible to obtain a $2\sqrt{m\log(n+k)}$ approximation in the general case of the problem. This remained the best known upper bound for Red-Blue Set Cover prior to our work. No non-trivial algorithms for MMSA_t were previously known for any $t \geq 4$.

On the hardness side, Dinur and Safra [10] showed that MMSA_3 is hard to approximate within a factor of $O(2^{\log^{1-\epsilon} m})$ where $\epsilon = 1/\log\log^c m$ for any constant $c < 1/2$, if $P \neq NP$. As was observed by Carr et al. [5], this implies a factor of $O(2^{\log^{1-\epsilon} m})$ hardness for Red-Blue Set Cover as well. The hardness result holds even for the special case of the problem where every set contains only one blue and two red elements.

Finally, Charikar et al. [7] gave a lower bound on a variant of MMSA in which the circuit depth t is not fixed. Assuming a variant of the Dense-vs-Random conjecture, they showed that for every $\varepsilon > 0$, the problem does not admit an $O(n^{1/2-\varepsilon})$ approximation, where n is the number of variables, and an $O(N^{1/3-\varepsilon})$ approximation, where N is the total number of gates and variables in the circuit.

Learning of Disjunctions. While algorithms for Red-Blue Set Cover return a disjunction with no error on positive examples, i.e., it covers all "blue" elements, it is straightforward to make those algorithms work for the case with two-sided error. A variant of the problem with a two-sided error is formally defined as *Positive–Negative Partial Set Cover* [15] where the author showed that a $f(m,n)$-approximation for Red-Blue Set Cover implies a $f(m+n,n)$-approximation for Positive-Negative Partial Set Cover. Our result also holds for Partial Red-Blue Set Cover and a c-approximation for Partial Red-Blue Set Cover can be used to output a c-approximate solution of Positive-Negative Partial Set Cover.

Awasthi et al. [4] designed an $O(n^{1/3+\alpha})$-approximation for any constant $\alpha > 0$. While the proposed algorithm of Awasthi et al. is an *agnostic learning* of disjunctions (i.e., the solution is not of form of disjunctions), employing an approximation algorithm of Red-Blue Set Cover, produces a disjunction as its output (i.e., the algorithms for Red-Blue Set Cover are *proper* learners).

Geometric Red-Blue Set Cover. The problem has been studied extensively in geometric models. Chan and Hu [6] studied the setting in which R and B are sets of points in $\mathbb{R}^2$ and $\mathcal{S}$ is a collection of unit squares. They proved that the problem still remains NP-hard in this restricted instance and presented a PTAS for this problem. Madireddy and Mudgal [13] designed an $O(1)$-approximation algorithm for another geometric variant of the problem, in which sets are unit disks. The problem has also been studied in higher dimensions with hyperplanes and axis-parallel objects as the sets [3, 14, 1].

1.2 Our Results

In this paper, we present new approximation algorithms for Red-Blue Set Cover, MMSA_4, and general MMSA_t. Additionally, we offer a new conditional hardness of approximation result for Red-Blue Set Cover. We also discuss the integrality gap of a basic SDP relaxation of MMSA_t strengthened by Sherali–Adams when $t \to \infty$.

We start by describing our result for Red-Blue Set Cover.

▶ **Theorem 3.** *There exists an $O(m^{1/3}\log^{4/3} n\log k)$-approximation algorithm for Red-Blue Set Cover where m is the number of sets, n is the number of red elements, and k is the number of blue elements.*

As we demonstrate later, our algorithm also works for the Partial Red-Blue Set Cover. Our approach partitions the instance into subinstances in which all sets have a bounded number of red elements, say between r and $2r$, and each red element appears in a bounded number of sets. Utilizing the properties of this partition, we show that we can always find a small collection of sets that preserves the right ratio of red to blue elements in order to make progress towards an $\tilde{O}(m^{1/3})$-approximation algorithm.[2] Then, by applying the algorithm iteratively until all blue elements are covered, we obtain the guarantee of Theorem 3. In each iteration, our analysis guarantees the feasibility of a local LP relaxation for which a simple randomized rounding obtains the required ratio of blue to red vertices.

Now we describe our results for the MMSA problem.

▶ **Theorem 4.** *There exists an $\tilde{O}(N^{1/3})$-approximation algorithm for $MMSA_4$, where N is the total number of gates and variables in the input instance.*

Our algorithm for MMSA_4 is inspired by our algorithm for MMSA_3, though due to the complexities of the problem, the algorithm is significantly more involved. In particular, there does not seem to be a natural preprocessing step analogous to the partition we apply for Red-Blue Set Cover, and so we need to rely on a higher-moment LP relaxation and a careful LP-based partition which is built into the algorithm.

▶ **Theorem 5.** *Let $t \geq 4$. Define $\delta = \frac{1}{3}\cdot 2^{3-\lceil t/2\rceil}$. There exists an $\tilde{O}(N^{1-\delta})$-approximation algorithm for $MMSA_t$ where N is the total number of gates and variables in the input instance.*

Our algorithm for general MMSA_t applies a recursion on the depth t, with our algorithms for Red-Blue Set Cover and MMSA_4 as the basis of the recursion. Each recursive step relies on an initially naive LP relaxation to which we add constraints as calls to the algorithm for smaller depth MMSA reveal new violated constraints.

We complement our upper bound for Red-Blue Set Cover with a hardness of approximation result.

▶ **Theorem 6.** *Assuming the Hypergraph Dense-vs-Random Conjecture, for every $\varepsilon > 0$, no polynomial-time algorithm achieves better than $O(m^{1/4-\varepsilon}/\log^2 k)$ approximation for Red-Blue Set Cover where m is the number of sets and k is the number of blue elements.*

To show the hardness, we present a reduction from Min k-Union to Red-Blue Set Cover that preserves the approximation up to a factor of polylog(k). Then, the hardness follows from the standard conjectured hardness of Min k-Union [9]. In our reduction, all elements of the given instance of Min k-Union are considered as the red elements in the constructed instance for Red-Blue Set Cover and we further complement each set with a sample size of $O(\log k)$ (with replacement) from a ground set of blue elements of size k. We prove that this reduction is approximation-preserving up to a factor of polylog(k).

[2] Here, we abuse the $\tilde{O}$ notation to hide polylog factors of n, k.

Organization. In Section 2, we restate Red-Blue Set Cover and introduce some notation. In Section 3, we present our algorithm for Red-Blue Set Cover. We adapt this algorithm for Partial Red-Blue Set Cover in Appendix A. Then, in Section 4 we give the algorithm for MMSA_t with $t \geq 5$. This algorithm relies on the algorithm for MMSA_4, which we describe later in Section 6. We present a reduction from Min k-Union to Red-Blue Set Cover, which yields a hardness of approximation result for Red-Blue Set Cover, in Section 5. The discussion on hardness of the general MMSA_t problem is deferred to the full version of the paper.

2 Preliminaries

To simplify the description and analysis of our approximation algorithm for Red-Blue Set Cover, we restate the problem in graph-theoretic terms. Essentially we restate the problem as MMSA_3. Specifically, we think of a Red-Blue Set Cover instance as a tripartite graph (B, J, R, E) in which all edges (E) are incident on J and either B or R. The vertices in J represent the set indices, and their neighbors in B (resp. R) represent the blue (resp. red) elements in these sets. Thus, our goal is to find a subset of the vertices in J that is a dominating set for B and has a minimum total number of neighbors in R. For short, we will denote the cardinality of these different vertex sets by $k = |B|$, $m = |J|$, and $n = |R|$.

Similarly, we think of a MMSA_4 instance as a tuple (B, J, R, S, E). Here, B, J, and R represent gates in the second, third, and fourth layers of the circuit (where layer i consists of the gates at distance $i-1$ from the root), respectively; S represents the variables; E represent edges between gates/variables. Combinatorially, the goal is to obtain a subset of J as above, along with a minimum dominating set in S for the red neighbors (in R) of our chosen subset of J.

Notation. We use $\Gamma(\cdot)$ to represent neighborhoods of vertices, and for a vertex set U, we use $\Gamma(U)$ to denote the union of neighborhoods of vertices in U, that is $\bigcup_{u \in U} \Gamma(u)$. We also consider restricted neighborhoods, which we denote by $\Gamma_T(u) := \Gamma(u) \cap T$ or $\Gamma_T(U) = \Gamma(U) \cap T$. We will refer to the cardinality of such a set, i.e. $|\Gamma_T(u)|$ as the T-*degree* of u.

▶ Remark 7. Note that, for every set index $j \in J$, the set $\Gamma(j)$ is simply the set S_j in the set system formulation of the problem, and the set $\Gamma_R(j)$ (resp. $\Gamma_B(j)$) is simply the subset of red (resp. blue) elements in the set with index j. Similarly, $\Gamma_J(i)$ consists of indices j representing those sets S_j that contain element i, for any $i \in R \cup B$.

For Red-Blue Set Cover algorithms, we introduce a natural notion of progress:

▶ **Definition 8.** *We say that an algorithm for Red-Blue Set Cover* makes progress towards an $O(A \cdot \log k)$-approximation *if, given an instance with an optimum solution containing* OPT *red elements, the algorithm finds a subset* $\hat{J} \subseteq J$ *such that* $\frac{|\Gamma_R(\hat{J})|}{|\Gamma_B(\hat{J})|} \leq A \cdot \frac{\text{OPT}}{|B|}$.

It is easy to see that if we have an algorithm which makes progress towards an A-approximation, then we can run this algorithm repeatedly (with decreasing $|B|$ parameter, where initially $|B| = k$) until we cover all blue elements and obtain an $O(A \cdot \log k)$-approximation. For brevity, all logarithms are implicitly base 2 unless otherwise specified, that is, we write $\log(\cdot) \equiv \log_2(\cdot)$.

3 Approximation Algorithm for Red-Blue Set Cover

We begin by excluding a small number of red elements, and binning the sets J into a small number of bins with uniform red-degree. For an $O(A)$-approximation, the goal will be to exclude at most $A \cdot \mathrm{OPT}$ red elements (we may guess the value of OPT by a simple linear or binary search). This is handled by the following lemma:

▶ **Lemma 9.** *There is a polynomial time algorithm, which, given an input (B, J, R, E) and parameter n_0, returns a set of at most $\log n$ pairs (J_α, R_α) with the following properties:*

- *The sets $J_\alpha \subseteq J$ partition the set J.*
- *The sets $R_\alpha \subseteq R$ form a nested collection of sets, and the smallest among them (i.e., their intersection) has cardinality at least $n - n_0$. That is, at most n_0 red elements are excluded by any of these sets.*
- *For every α there is some r_α such that every set $j \in J_\alpha$ has R_α-degree (or restricted red set size) $|\Gamma_{R_\alpha}(j)| \in [r_\alpha, 2r_\alpha]$,*
- *and for every α, every red element $i \in R_\alpha$ has J_α-degree at most (that is, the number of red sets in J_α that contain i) $|\Gamma_{J_\alpha}(i)| \le \frac{2mr_\alpha \log n}{n_0}$.*

Proof. Consider the following algorithm:

- Let r be the maximum red-degree (i.e., $\max_{j\in J} \deg_R(j)$).
- Repeat the following while $J \neq \emptyset$:
  - Delete the top $n_0/\log n$ J-degree red elements from R, along with their incident edges.
  - After this deletion, let $J' = \{j \in J \mid \deg_R(j) \in [r/2, r]\}$.
  - If J' is non-empty, add the current pair (J', R) to the list of output pairs (excluding all elements deleted from R so far).
  - Remove the sets in J' from J (along with incident edges) and let $r \leftarrow r/2$.

By the decrease in r, it is easy to see that this partitions J into at most $\log n$ sets (or more specifically, log of the initial maximum red set size, $\max_{j\in J} \deg_R(j)$). Also note that at the beginning of each iteration, all red sets have size at most r, and so there are at most mr edges to R, and the top $n_0/\log n$ J-degree red elements will have average degree (and in particular minimum degree) at most $mr/(n_0/\log n)$. Thus, after removing these red elements, all remaining red elements will have J-degree (and in particular J'-degree) at most the required bound of $\frac{2mr_\alpha \log n}{n_0}$ where $r_\alpha = r/2$. Finally, since there at most $\log n$ iterations, the total number of red elements removed across all iterations is at most n_0. ◀

Our algorithm works in iterations, where at every iteration, some subset of blue elements is covered and removed from B. However, nothing is removed from J or R. Thus the above lemma applies throughout the algorithm. Note that for an optimum solution $J_{\mathrm{OPT}} \subseteq J$, for at least one of the subsets J_α in the above partition, the sets in $J_{\mathrm{OPT}} \cap J_\alpha$ must cover at least a $(1/\log n)$-fraction of B. Thus, to achieve an $O(A)$ approximation, it suffices to apply the above lemma with parameter $n_0 = \mathrm{OPT} \cdot A$, and repeatedly make progress towards an A-approximation within one of the subgraphs induced on (B, J_α, R_α). We will only pay at most another $\mathrm{OPT} \cdot A$ in the final analysis by restricting our attention to these subinstances.

Let us fix some optimum solution $J_{\mathrm{OPT}} \subseteq J$ in advance. For any α in the above partition, let $J^\alpha_{\mathrm{OPT}} = J_\alpha \cap J_{\mathrm{OPT}}$ be the collection of sets in J_α that also belong to our optimum solution, and let $B_\alpha = \Gamma_B(J^\alpha_{\mathrm{OPT}})$ be the blue elements covered by the sets in J^α_{OPT}. Note that every blue element must belong by the feasibility of J_{OPT} to at least one B_α. In this context we can show the following:

▶ **Lemma 10.** *For any α in the partition described in Lemma 9, there exists a red element $i_0 \in R_\alpha$ such that its optimum J_α-restricted neighbors $\Gamma_{J^\alpha_{\mathrm{OPT}}}(i_0)$ cover at least $|B_\alpha|r_\alpha/\mathrm{OPT}$ blue elements.*

Proof. Consider the following subgraph of $(B_\alpha, J^\alpha_{\mathrm{OPT}}, E(B_\alpha, J^\alpha_{\mathrm{OPT}}))$. For every blue element $\ell \in B_\alpha$, retain exactly one edge to J^α_{OPT}. Let $\hat{F}$ be this set of edges.

Thus the blue elements B_α have at least $|B_\alpha|r_\alpha$ paths through $\hat{F} \times E(J^\alpha_{\mathrm{OPT}}, R_\alpha)$ to the red neighbors of J^α_{OPT} in R_α. Since there are at most OPT such red neighbors, at least one of them, say $i_0 \in \Gamma_{R_\alpha}(J^\alpha_{\mathrm{OPT}})$, must be involved in at least a $1/\mathrm{OPT}$ fraction of these paths. That is, at least $|B_\alpha|r_\alpha/\mathrm{OPT}$ paths. Since the $\hat{F}$-neighborhoods of the vertices in J^α_{OPT} are disjoint (by construction), these paths end in distinct blue elements, thus, at least $|B_\alpha|r_\alpha/\mathrm{OPT}$ elements in B_α. ◀

Of course, we cannot know which red element will have this property, but the algorithm can try all elements and run the remainder of the algorithm on each one. Now, given a red element $i_0 \in R_\alpha$, our algorithm proceeds as follows: Begin by solving the following LP.

$$\max \sum_{\ell \in B} z_\ell \tag{1}$$

$$\sum_{i \in R_\alpha} y_i \leq \mathrm{OPT} \tag{2}$$

$$0 \leq z_\ell \leq \min\{1, \sum_{j \in \Gamma_{J_\alpha}(i_0) \cap \Gamma_{J_\alpha}(\ell)} x_j\} \qquad \forall \ell \in B \tag{3}$$

$$0 \leq x_j \leq y_i \leq 1 \qquad \forall j \in \Gamma_{J_\alpha}(i_0), i \in \Gamma_{R_\alpha}(j) \tag{4}$$

In the intended solution, x_j is the indicator for the event that $j \in \Gamma_{J^\alpha_{\mathrm{OPT}}}(i_0)$, y_i is the indicator variable for the event that red vertex i is in the union of red sets indexed by $\Gamma_{J^\alpha_{\mathrm{OPT}}}(i_0)$ (and therefore in the optimum solution), and z_ℓ is the indicator variable for the event that the blue vertex $\ell \in B$ is is covered by some set in $\Gamma_{J^\alpha_{\mathrm{OPT}}}(i_0)$. This LP is always feasible (say, by setting all variables to 0), though since there are at most $\log n$ subinstances in the partition, for at least one α we must have $|B_\alpha| \geq |B|/\log n$, and then by Lemma 10, there is also some choice of $i_0 \in R_\alpha$ for which the objective function satisfies

$$\sum_{\ell \in B} z_\ell \geq \frac{|B_\alpha|r_\alpha}{\mathrm{OPT}} \geq \frac{|B|r_\alpha}{\mathrm{OPT} \cdot \log n}. \tag{5}$$

The algorithm will choose α and i_0 that maximize the rescaled objective function $\sum_{\ell \in B} z_\ell/r_\alpha$, guaranteeing this bound. Finally, at this point, we perform a simple randomized rounding, choosing every set $j \in \Gamma_{J_\alpha}(i_0)$ independently with probability x_j. The entire algorithm is described in Algorithm 1.

Now let $J^* \subseteq J_\alpha$ be the collection of sets added by this step in the algorithm. Let us analyze the number of blue elements covered by J^* and the number of red elements added to the solution. First, noting that this LP acts as a max coverage relaxation for blue elements, the expected number of blue elements covered will be at least $(1-1/e)\frac{|B|r_\alpha}{\mathrm{OPT} \cdot \log n}$, by the standard analysis and the bound (5).

Now let us bound the number of red elements added. Let $J_+ = \left\{ j \in \Gamma_{J_\alpha}(i_0) \;\middle|\; x_j \geq \frac{\mathrm{OPT}}{r_\alpha \hat{A}} \right\}$ for a value of $\hat{A}$ to be determined later. By Constraint (4), every red neighbor $i \in \Gamma_{R_\alpha}(J_+)$ will also have $y_i \geq \mathrm{OPT}/(r_\alpha \hat{A})$, and so by Constraint (2), there can be at most $r_\alpha \hat{A}$ such neighbors. On the other hand, the expected number of red elements added by the remaining sets $j \in J^* \setminus J_+$ is bounded by

Algorithm 1 Approximation Algorithm for Blue-Red Set Cover.

Input: B, J, R, E
guess OPT ▷ e.g. using binary search
$J_{\text{ALG}} = \emptyset$ ▷ J_{ALG} stores the current solution
find decomposition $\{(J_\alpha, R_\alpha)\}_\alpha$ as in Lemma 9, with $n_0 = \text{OPT} \cdot m^{1/3} \log^{4/3} n \log^2 k$.
while $B \neq \emptyset$ **do**
  for all α and $i_0 \in R_\alpha$ **do**
    Solve LP (1)-(4). Let $\text{LP}(\alpha, i_0)$ be its objective value
  end for
  choose the value of α and i_0 which maximizes $\text{LP}(\alpha, i_0)/r_\alpha$
  let x, y, z be an optimal solution for this LP
  use solution x and the method of conditional expectations to find $J^* \subseteq J_\alpha$
   s.t. $|\Gamma_{R_\alpha}(J^*)|/|\Gamma_B(J^*)| \leq O(1) \cdot m^{1/3} \log^{4/3} n \cdot \text{OPT}/|B|$ ▷ see the proof for details
  let $J_{\text{ALG}} = J_{\text{ALG}} \cup J^*$
  let $B = B \setminus \Gamma_B(J^*)$
  remove edges incident to deleted vertices from E
end while
return J_{ALG}

$$
\begin{aligned}
\mathbb{E}\left[\left|\bigcup_{j \in J^* \setminus J_+} \Gamma_{R_\alpha}(j)\right|\right] &\leq 2r_\alpha \mathbb{E}\left[|J^* \setminus J_+|\right] && \text{by } R_\alpha\text{-degree bounds for } j \in J_\alpha \\
&= 2r_\alpha \cdot \sum_{j \in \Gamma_{J_\alpha}(i_0) \setminus J_+} x_j \\
&\leq 2r_\alpha \cdot \frac{\text{OPT}}{r_\alpha \hat{A}} |\Gamma_{J_\alpha}(i_0) \setminus J_+| && \text{by definition of } J_+ \\
&\leq \frac{2\text{OPT}}{\hat{A}} \cdot \frac{2mr_\alpha \log n}{\text{OPT} \cdot \hat{A}} && \text{by } J_\alpha\text{-degree bounds for } i \in R_\alpha \\
&= \frac{4mr_\alpha \log n}{\hat{A}^2}
\end{aligned}
$$

These two bounds are equal when $r_\alpha \hat{A} = 4mr_\alpha \log n/\hat{A}^2$, that is, when $\hat{A} = (4m \log n)^{1/3}$, giving us a total bound on the expected number of red elements added in this step of

$$
\mathbb{E}[|\Gamma_{R_\alpha}(J^*)|] \leq 2r_\alpha(4m \log n)^{1/3} \leq \frac{2r_\alpha(4m \log n)^{1/3}}{(1 - 1/e)|B|r_\alpha/(\text{OPT} \cdot \log n)} \cdot \mathbb{E}[|\Gamma_B(J^*)|].
$$

Thus, $\mathbb{E}[|\Gamma_{R_\alpha}(J^*)|] - \frac{2(4m \log^4 n)^{1/3}\text{OPT}}{(1-1/e)|B|} \cdot \mathbb{E}|\Gamma_B(J^*)| \leq 0$. Using the method of conditional expectations, we can derandomize the algorithm and find J^* with a non-empty blue neighbor set such that $\frac{|\Gamma_{R_\alpha}(J^*)|}{|\Gamma_B(J^*)|} \leq O(1) \cdot m^{1/3} \log^{4/3} n \cdot \frac{\text{OPT}}{|B|}$. Thus, we make progress (according to Definition 8) towards an approximation guarantee of $\tilde{A} \cdot k$ for $\tilde{A} = O\left(m^{1/3} \log^{4/3} n\right)$, which, as noted, ultimately gives us the same approximation guarantee for Red-Blue Set Cover, proving Theorem 3.

4 Approximating MMSA$_t$ for $t \geq 5$

We now turn to the general problem of approximating MMSA$_t$ for arbitrarily large (but fixed) t. We will build on our approximation algorithm for MMSA$_4$ (described in Section 6) by recursively calling approximation algorithms for the problem with smaller values of t, and using the result of this approximation as a separation oracle in certain cases.

We will denote the total size of our input by N, and we will denote our approximation factor for MMSA_t by A_t. We will only describe an algorithm for even depth. There is a very slightly simpler but quite similar algorithm for odd depth, however the guarantee we are able to achieve for MMSA_{2t-1} is nearly the same as for MMSA_{2t} (up to an $O(\log N)$ factor). Since MMSA_{2t-1} is essentially a special case of MMSA_{2t}, we focus only on even levels.

▶ **Lemma 11.** *For $t \geq 2$, if MMSA_{2t} can be approximated to within a factor of A_{2t}, then we can approximate MMSA_{2t+2} (and thus MMSA_{2t+1}) to within $O(\sqrt{N \cdot A_{2t}} \log N)$.*

Proof. Denote our input as a layered graph with vertex layers $V_1, \ldots, V_{2t+2}$. Ideally, we would like to discard any vertex $j \in V_{2t}$ such that covering its neighbors $\Gamma_{2t+1}(j)$ requires more than OPT vertices in V_{2t+2}, however, checking this precisely requires solving Set Cover. Instead, we discard any vertex $j \in V_{2t}$ for which the smallest *fractional* set cover[3] in V_{2t+2} of its neighbors $\Gamma_{V_{2t+1}}(j)$ has value greater than OPT. Such vertices cannot be included without incurring cost greater than OPT and so we know they do not participate in any optimum solution. We begin with the following basic LP:

$$\sum_{h \in V_{2t+2}} w_h \leq \mathrm{OPT} \tag{6}$$

$$y_i \leq \sum_{h \in \Gamma_{2t+2}(i)} w_h \qquad \forall i \in V_{2t+1} \tag{7}$$

$$x_j \leq y_i \qquad \forall j \in V_{2t}, i \in \Gamma_{V_{2t+1}}(j) \tag{8}$$

$$x_j, y_i, w_h \in [0,1] \qquad \forall j \in V_{2t} \forall i \in V_{2t+1} \forall h \in V_{2t+2} \tag{9}$$

Note that, as stated, this LP is trivial. Indeed, in the absence of any additional constraints, the all-zero solution is feasible. However, we will add new violated constraints as the algorithm proceeds.

Given a solution to the above linear program, our algorithm for MMSA_{2t+2} is as follows:

- Let $V_{2t}^+ = \{j \in V_{2t} \mid x_j \geq 2(1+\ln N)/A_{2t+2}\}$. Add these vertices to the solution.
- Let $V_{2t+2}^+ = \Gamma_{V_{2t+2}}(\Gamma_{V_{2t+1}}(V_{2t}^+))$ be the neighbors-of-neighbors of V_{2t}^+.
- Apply a greedy $(1+\ln N)$-approximation for Set Cover to obtain a set cover (in V_{2t+2}) for $\Gamma_{V_{2t+1}}(V_{2t}^+)$, and add this set cover to the solution as well.
- Create an instance of MMSA_{2t} by removing layers V_{2t+1}, V_{2t+2}, all vertices in V_{2t}^+, as well as their neighbors in V_{2t-1}, that is, $\Gamma_{V_{2t-1}}(V_{2t}^+)$, as these are already covered by V_{2t}^+.
- Apply an A_{2t}-approximation algorithm for MMSA_{2t} to this instance, and let $U_{\mathrm{ALG}} \subseteq V_{2t} \setminus V_{2t}^+$ be the result (or at least the portion belonging to layer $2t$).
- If $|U_{\mathrm{ALG}}| \leq A_{2t+2}/(2+2\ln N)$, add the vertices in U_{ALG} to the solution, as well as a greedy set cover (in V_{2t+2}) for the neighborhood $\Gamma_{V_{2t+1}}(U_{\mathrm{ALG}})$.
- Otherwise (if $|U_{\mathrm{ALG}}| > A_{2t+2}/(2+2\ln N)$), continue the Ellipsoid algorithm using the new violated constraint

$$\sum_{j \in V_{2t} \setminus V_{2t}^+} x_j \geq \left\lfloor \frac{A_{2t+2}}{2(1+\ln N)A_{2t}} \right\rfloor + 1, \tag{10}$$

  and restart the algorithm (discarding the previous solution) using the new LP solution.

[3] That is, $\min\left\{\sum_{h\in S} z_h \;\middle|\; \forall i \in \Gamma_{V_{2t+1}}(j) : \sum_{h \in \Gamma_{V_{2t+2}}(i)} z_h \geq 1;\ \forall h \in S : z_h \geq 0\right\}$.

Let us now analyze this algorithm. By (8), we know that all neighbors $i \in V_{2t+1}$ of V_{2t}^+ have LP value $y_i \geq 2(1+\ln N)/A_{2t+2}$. Thus, by (7), if for every vertex $h \in V_{2t+2}$ we define $w_h^+ = w_h \cdot A_{2t+2}/(2+2\ln N)$, then this is a fractional Set Cover for the V_{2t+1}-neighborhood $\Gamma_{V_{2t+1}}(V_{2t}^+)$, and by (6) it has total fractional value at most $\mathrm{OPT} \cdot A_{2t+2}/(2+2\ln N)$. Thus, the greedy Set Cover $(1+\ln N)$-approximation algorithm will cover this neighborhood using at most $\mathrm{OPT} \cdot A_{2t+2}/2$ vertices in V_{2t+2}. After this step, we may add at most $\mathrm{OPT} \cdot A_{2t+2}/2$ additional vertices in V_{2t+2} to our solution to obtain an A_{2t+2}-approximation.

Now, suppose our MMSA_{2t} approximation returns a set U_{ALG} of cardinality $|U_{\mathrm{ALG}}| \leq A_{2t+2}/(2+2\ln N)$. Clearly, adding to our solution the vertices of U_{ALG} and a V_{2t+2}-Set Cover for its neighborhood $\Gamma_{V_{2t+1}}(U_{\mathrm{ALG}})$ gives a feasible solution to our MMSA_{2t+2} instance. Moreover, since by our preprocessing, the neighborhood $\Gamma_{V_{2t+1}}(j)$ of every $j \in U_{\mathrm{ALG}}$ has a fractional Set Cover in V_{2t+2} of value at most OPT, it follows that the union of all these neighborhoods, that is $\Gamma_{V_{2t+1}}(U_{\mathrm{ALG}})$, has a fractional set cover in V_{2t+2} of value at most $\mathrm{OPT} \cdot |U_{\mathrm{ALG}}| \leq \mathrm{OPT} \cdot A_{2t+2}/(2+2\ln N)$. And so applying a greedy Set Cover algorithm for the neighborhood $\Gamma_{V_{2t+1}}(U_{\mathrm{ALG}})$ contributes at most an additional $\mathrm{OPT} \cdot A_{2t+2}/2$ vertices in V_{2t+2} to our solution, as required.

Finally, let us examine the validity of the final step (the separation oracle). If the A_{2t}-approximation for MMSA_{2t} was not able to find a solution of size at most $A_{2t+2}/(2+2\ln N)$, then by definition, the value of any solution to our MMSA_{2t} instance must be greater than $A_{2t+2}/((2+2\ln N)A_{2t})$. This is a subinstance of our original instance, so any solution to our original MMSA_{2t+2} instance must also contain more than $A_{2t+2}/((2+2\ln N)A_{2t})$ vertices in V_{2t}. Thus, Constraint (10) is valid for any optimum solution. But when is it violated?

By definition of V_{2t}^+, the current total LP value of $V_{2t} \setminus V_{2t}^+$ is at most $2(1+\ln N)N/A_{2t+2}$. And so the current LP solution violates (10) if

$$\frac{2(1+\ln N)N}{A_{2t+2}} \leq \frac{A_{2t+2}}{2(1+\ln N)A_{2t}} \qquad \Longleftrightarrow \qquad A_{2t+2} \geq 2(1+\ln N)\sqrt{N \cdot A_{2t}}.$$

Thus, we can obtain an approximation factor of $A_{2t+2} = O(\sqrt{N \cdot A_{2t}} \log N)$ as claimed. ◀

Thus, by induction on t, with the guarantee of Theorem 4 for MMSA_4 as the basis of the induction, and Lemma 11 for the inductive steps, we get a general approximation algorithm for MMSA_t with approximation ratio $O\left(N^{1-\frac{1}{3}2^{3-\lceil t/2 \rceil}} \cdot (\log N)^{2+O(2^{-t/2})}\right)$.

5 Reduction from Min k-Union to Red-Blue Set Cover

In this section, we first present a reduction from Min k-Union to Red-Blue Set Cover and then prove a hardness result for Red-Blue Set Cover. We start with formally defining the Min k-Union problem.

▶ **Definition 12** (Min k-Union). *In the Min k-Union problem, we are given a set X of size n, a family $\mathcal{S}$ of m sets $S_1, \ldots, S_m$, and an integer parameter $k \geq 1$. The goal is to choose k sets $S_{i_1}, \ldots, S_{i_k}$ so as to minimize the cost $\left|\bigcup_{t=1}^{k} S_{i_t}\right|$. We will denote the cost of the optimal solution by $\mathrm{OPT}_{MU}(X, \mathcal{S}, k)$.*

Note that Min k-Union resembles the Red-Blue Set Cover Cover problem: in both problems, the goal is to choose some subsets $S_{i_1}, \ldots, S_{i_r}$ from a given family $\mathcal{S}$ so as to minimize the number of elements or red elements in their union. Importantly, however, the feasibility requirements on the chosen subsets $S_{i_1}, \ldots, S_{i_r}$ are different in Red-Blue Set Cover Cover and Min k-Union; in the former, we require that the chosen sets cover all k blue points but in the latter, we simply require that the number of chosen sets be k. Despite this difference, we show that there is a simple reduction from Min k-Union to Red-Blue Set Cover.

▷ Claim 13. There is a randomized polynomial-time reduction from Min k-Union to Red-Blue Set Cover that given an instance $\mathcal{I} = (X, \mathcal{S}, k)$ of Min k-Union returns an instance $\mathcal{I}' = (R, B, \{S'_i\}_{i\in[m]})$ of Red-Blue Set Cover satisfying the following two properties:

1. If $S'_{j_1}, \dots, S'_{j_r}$ is a feasible solution for $\mathcal{I}'$ then $k' \le r$ and the cost of solution $S_{j_1}, \dots, S_{j_{k'}}$ for Min k'-Union where $k' = \lfloor k/\ell \rfloor$ and $\ell = \lceil \log_e k \rceil + 1$ does not exceed that of $S'_{j_1}, \dots, S'_{j_r}$ for Red-Blue Set Cover:
$$\mathrm{cost}_{MU}(S_{j_1}, \dots, S_{j_{k'}}) \equiv \left| \bigcup_{t=1}^{k'} S_{i_t} \right| \le \left| \bigcup_{t=1}^{r} (S'_{i_t} \cap R) \right| = \mathrm{cost}_{RB}(S'_{j_1}, \dots, S'_{j_r}).$$
This is true always no matter what random choices the reduction makes.
2. $\mathrm{OPT}_{RB}(R, B, \{S'_i\}_i) \le \mathrm{OPT}_{MU}(X, \mathcal{S}, k)$ with probability at least $1 - 1/e$.

Proof. We define instance $\mathcal{I}'$ as follows. Let $R = X$ and $B = [k]$. For every $i \in [m]$, let $R_i = S_i$; B_i be a set of ℓ elements randomly sampled from $[k]$ with replacement, and $S'_i = R_i \cup B_i$. All random choices are independent. Now we verify that this reduction satisfies the required properties.

Consider a feasible solution $S'_{j_1}, \dots, S'_{j_r}$ for $\mathcal{I}'$. Since this solution is feasible, $\cup_{t=1}^{r} B_t = B$. Now $|B_t| \le \ell$ and thus $r \ge |B|/\ell = k/\ell \ge k'$, as required. Further,
$$\mathrm{cost}_{MU}(S_{j_1}, \dots, S_{j_{k'}}) \equiv \left| \bigcup_{t=1}^{k'} S_{j_t} \right| = \left| \bigcup_{t=1}^{k'} R_{j_t} \right| \le \left| \bigcup_{t=1}^{r} R_{j_t} \right| \equiv \mathrm{cost}_{RB}(S'_{j_1}, \dots, S'_{j_r}).$$
We have verified that item 1 holds. Now, let $S_{i_1}, \dots, S_{i_k}$ be an optimal solution for $\mathcal{I}$. We claim that $S'_{i_1}, \dots, S'_{i_k}$ is a feasible solution for $\mathcal{I}'$ with probability at least $1 - 1/e$. To verify this claim, we need to lower bound the probability that $B_{i_1} \cup \dots \cup B_{i_k} = B$. Indeed, set $B_{i_1} \cup \dots \cup B_{i_k}$ consists of $k\ell$ elements sampled from B with replacement. The probability that a given element $b \in B$ is not in $B_{i_1} \cup \dots \cup B_{i_k}$ is at most $(1 - 1/k)^{k\ell} \le e^{-\ell} \le \frac{1}{ek}$. By the union bound, the probability that there is some $b \in B \setminus (B_{i_1} \cup \dots \cup B_{i_k})$ is at most $k \times \frac{1}{ek} = \frac{1}{e}$. Thus, there is no such b with probability at least $1 - 1/e$ and consequently $B_{i_1} \cup \dots \cup B_{i_k} = B$. In that case, the cost of solution $S'_{i_1}, \dots, S'_{i_k}$ for Red-Blue Set Cover equals $\left| \bigcup_{i=1}^{k} R'_{i_t} \right| = \left| \bigcup_{i=1}^{k} S_{i_t} \right|$, the cost of the optimal solution for Min k-Union. ◁

▶ **Corollary 14.** *Assume that there is an $\alpha(m, n)$ approximation algorithm $\mathcal{A}$ for Red-Blue Set Cover (where α is a non-decreasing function of m and n).Then there exists a randomized polynomial-time algorithm $\mathcal{B}$ for Min k-Union that finds k' sets $S_{i_1}, \dots, S_{i_{k'}}$ such that*
$$\left| \bigcup_{t=1}^{k'} S_{i_t} \right| \le \alpha(m, n) \mathrm{OPT}_{MU}(X, \mathcal{S}, k).$$
The failure probability is at most $1/n$.

Proof. We simply apply the reduction to the input instance of Min k-Union and then solve the obtained instance of Red-Blue Set Cover using algorithm $\mathcal{A}$. To make sure that the failure probability is at most $1/n$, we repeat this procedure $\lceil \log_e n \rceil$ times and output the best of the solutions we found. ◀

▶ **Theorem 15.** *Assume that there is an $\alpha(m,n)$ approximation algorithm $\mathcal{A}$ for Red-Blue Set Cover (where α is a non-decreasing function of m and n). Then there exists an $O(\log^2 k)\alpha(m,n)$ approximation algorithm for Min k-Union.*

Proof. Our algorithm iteratively uses algorithm $\mathcal{B}$ from the corollary to find an approximate solution. First, it runs $\mathcal{B}$ on the input instance and gets $k_1 = k'$ sets. Then it removes the sets it found from the instance and reduces the parameter k to $k - k_1$. Then the algorithm runs $\mathcal{B}$ on the obtained instance and gets k_2 sets. It again removes the obtained sets and reduces k to $k - k_1 - k_2$ (here k is the original value of k). It repeats these steps over and over until it finds k sets in total. That is, $k_1 + \cdots + k_T = k$ where T is the number of iterations the algorithm performs.

Observe that each of the instances of Min k-Union constructed in this process has cost at most $\mathrm{OPT}_{MU}(X, \mathcal{S}, k)$. Indeed, consider the subinstance $\mathcal{I}_{t+1}$ we solve at iteration $t+1$. Consider k sets that form an optimal solution for $(X, \mathcal{S}, k)$. At most $k_1 + \cdots + k_t$ of them have been removed from $\mathcal{I}_{t+1}$ and thus at least $k - k_1 - \cdots - k_t$ are still present in $\mathcal{I}_{t+1}$. Let us arbitrarily choose $k - k_1 - \cdots - k_t$ sets among them. The chosen sets form a feasible solution for $\mathcal{I}_{t+1}$ of cost at most $\mathrm{OPT}_{MU}(X, \mathcal{S}, k)$.

Thus, the cost of a partial solution we find at each iteration t is at most $\alpha(m,n) \cdot \mathrm{OPT}_{MU}(X, \mathcal{S}, k)$. The total cost is at most $\alpha(m,n) \cdot T \cdot \mathrm{OPT}_{MU}(X, \mathcal{S}, k)$. It remains to show that $T \leq O(\log^2 k)$. We observe that the value of k reduces by a factor at least $1 - 1/\ell$ in each iteration, thus after t iterations it is at most $(1 - 1/\ell)^t k$. We conclude that the total number of iterations T is at most $O(\ell \log k) = O(\log^2 k)$, as desired. ◀

Now we obtain a conditional hardness result for Reb-Blue Set Cover from a corollary from the Hypergraph Dense-vs-Random Conjecture.

▶ **Corollary 16** (Chlamtáč et al. [9]). *Assuming the Hypergraph Dense-vs-Random Conjecture, for every $\varepsilon > 0$, no polynomial-time algorithm for Min k-Union achieves better than $\Omega(m^{1/4-\varepsilon})$ approximation.*

Theorem 6 immediately follows.

6 Approximation Algorithm for MMSA$_4$

Consider an instance (B, J, R, S, E) of MMSA$_4$. As we did for Red-Blue Set Cover, we will focus on making progress towards a good approximation. Due to space constraints, we have omitted most proofs for statements in this section. Complete proofs can be found in the full version of the paper.

▶ **Definition 17.** *We say that an algorithm for MMSA$_4$* makes progress towards an $O(A)$-approximation *if, given an instance with an optimum solution containing at most* OPT *vertices in S, the algorithm finds a subset $\hat{J} \subseteq J$ and a subset $\hat{S} \subseteq S$ such that $\Gamma_R(\hat{J}) \subseteq \Gamma_R(\hat{S})$ (a valid partial solution) and $\frac{|\hat{S}|}{|\Gamma_B(\hat{J})|} \leq A \cdot \frac{\mathrm{OPT}}{|B|}$.*

As before, it is easy to see that given such an algorithm, we can run such an algorithm repeatedly to obtain an actual $\tilde{O}(A)$ approximation for MMSA$_4$. In fact, in the rest of this section we will *only* discuss an algorithm which makes progress towards an $O(A)$-approximation.

For the sake of formulating an LP relaxation with a high degree of uniformity, we will actually focus on a partial solution which covers a large fraction of blue elements in a uniform manner:

▶ **Lemma 18.** *For any cover $J_0 \subseteq J$ of the blue elements B, there exist subsets $J' \subseteq J_0$ and $B' \subseteq B$ and a parameter $\Delta > 0$ with the following properties:*

- *Every vertex $j \in J'$ has B'-degree in the range $\deg_{B'}(j) \in [\Delta, 2\Delta]$.*
- *Every blue element $\ell \in B'$ has at least one neighbor in J'_Δ and at most $2e\ln(2k)$ neighbors.*
- *We have the cardinality bound $|B'| \geq |B|/(\log k \log m)$.*

Simplifying Assumptions. We can make the following assumptions which will be useful in the analysis of our algorithm. First, we may assume that for every $j \in J$, the red neighborhood $\Gamma_R(j)$ has a fractional set cover in S of weight at most OPT. That is, the standard LP relaxation for covering $\Gamma_R(j)$ using S has optimum value at most OPT. If we have guessed the correct value of OPT, then we know that no $j \in J$ whose red neighborhood cannot be covered with cost OPT can participate in an optimum solution, and can therefore be discarded. We may also assume that for some $\varepsilon > 0$, the value Δ above is at most k/m^ε. The reason is that otherwise, the blue elements B' can be covered with at most $\tilde{O}(m^\varepsilon)$ vertices in J, and we know that for each of these, its red neighborhood can be covered by a set of size $\tilde{O}(\text{OPT})$ in S, and thus we can make progress towards an $\tilde{O}(m^\varepsilon)$ approximation.

Guessing the value of $\Delta \in [k]$ above and the value of the optimum OPT, we can write the following LP relaxation:

$$\sum_{h \in S} w_h \leq \text{OPT} \tag{11}$$

$$\sum_{\ell \in B} z_\ell \geq |B|/(\log k \log m) \tag{12}$$

$$z_\ell \leq \sum_{j \in \Gamma_J(\ell)} x_j^\ell \leq 2e\ln(2k) z_\ell \qquad \forall \ell \in B \tag{13}$$

$$\Delta x_j \leq \sum_{\ell \in \Gamma_B(j)} x_j^\ell \leq 2\Delta x_j \qquad \forall j \in J \tag{14}$$

$$0 \leq x_j^\ell \leq x_j, z_\ell \leq 1 \qquad \forall \ell \in B \forall j \in J \tag{15}$$

$$\sum_{h \in \Gamma_S(i)} w_h \geq y_i \qquad \forall i \in R \tag{16}$$

$$0 \leq x_j \leq y_i \qquad \forall (j,i) \in E(J,R) \tag{17}$$

We further strengthen this LP by partially lifting the above constraints. Specifically, for every $a \in J \cup S$, $j \in J$, $h \in S$, $i \in R$, and $\ell \in B$ we introduce variables $X_h^{(a)}, X_\ell^{(a)}, X_{\ell,j}^{(a)}, X_j^{(a)}, X_i^{(a)}$, and lift all the above constraints accordingly. For a precise definition, see Appendix B. For any $j \in J$ such that $x_j > 0$ or $h \in S$ such that $w_h > 0$, we will denote the "conditioned" variables by $\hat{w}_h^{(j)} = X_h^{(j)}/x_j$, $\hat{z}_\ell^{(h)} = X_\ell^{(h)}/w_h$, etc.

▶ Remark 19. The above linear program is a relaxation for the partial solution given by Lemma 18. Specifically, given an optimal solution $(J_{\text{OPT}}, S_{\text{OPT}})$, applying the lemma to $J_0 = J_{\text{OPT}}$, we have the following feasible solution: Set the variables z_ℓ and x_j to be indicators for B' and J' as in the lemma, respectively, and the variables x_j^ℓ to be indicators for $J' \times B'$. Set the the variables w_h to be indicators for S_{OPT}, and the variables y_i to be indicators for the red neighbors $\Gamma_R(J')$ of J'.

Let us examine some useful properties of this relaxation. First of all, we note that it approximately determines the total LP value of J (since the LP assigns total LP value $\tilde{\Theta}(|B|)$ to B):

▷ Claim 20. Any solution satisfying constraints (13)-(15), has total LP weight in J bounded by $\frac{1}{2\Delta}\sum_{\ell\in B} z_\ell \le \sum_{j\in J} x_j \le \frac{2e\ln(2k)}{\Delta}\sum_{\ell\in B} z_\ell$.

These constraints also determine a useful combinatorial property: in any feasible solution, the number of blue neighbors a subset of J has is (at least) proportional to the LP value of that set.

▷ Claim 21. For any solution satisfying constraints (13)-(15), and any subset of vertices $\hat{J} \subseteq J$, the number of blue neighbors of $\hat{J}$ is bounded from below by:

$$|\Gamma_B(\hat{J})| \ge \frac{1}{4e\ln(2k)\log k\log m}\cdot\frac{x(\hat{J})}{x(J)}\cdot|B|.$$

A fractional variant of the above covering property for the blue vertices is the following:

▷ Claim 22. For any solution satisfying constraints (13)-(15), and any subset of vertices $\hat{J} \subseteq J$, at least $\varepsilon|B|$ vertices $\ell \in B$ satisfy $\sum_{j\in\Gamma_{\hat{J}}(\ell)} x_j^\ell \ge \frac{1}{4\log k\log m}\cdot\frac{x(\hat{J})}{x(J)}$, where $\varepsilon = \frac{1}{8e\ln(2k)\log k\log m}\cdot\frac{x(\hat{J})}{x(J)}$.

Let us now analyze the approximation guarantee of Algorithm 2. We begin by stating simple lower bounds on the total LP value of the set J_0 as well as the vertices in the set.

▶ **Lemma 23.** *The set J_0 defined in Algorithm 2 has LP value at least $x(J)/(2\log m)$ and the lower bound on the individual LP values in the set is bounded by $x_0 \ge 1/m$.*

Next, we examine the bucketing of neighbors in S, and give a lower bound on the number of vertices in these bucketed sets.

▶ **Lemma 24.** *In Algorithm 2, for every vertex $j \in J_0$, and every red neighbor $i \in \Gamma_R(j)$, the bucketed set of neighbors $\hat{\Gamma}_j(i)$ of i has cardinality bounded from below by $|\hat{\Gamma}_j(i)| \ge 1/(6\beta_{ji}\log|S|\log(|S|^2 m))$.*

Proof. Fix vertices $j \in J_0$ and $i \in \Gamma_R(j)$. Let us begin by examining our choice of β_{ji}. Note that lifting Constraint 17, we get $x_j = X_j^{(j)} \le X_i^{(j)} (\le x_j)$, and so $\hat{y}^{(j)} = X_i^{(j)}/x_j = 1$. Lifting Constraint (16), we thus get $\sum_{h\in\Gamma_S(i)} \hat{w}_h^{(j)} \ge 1$. Note that the total LP weight of the set $S'_{ji} = \{h \in \Gamma_S(i) \mid \hat{w}_h^{(j)} \le 1/|S|^2\}$ is at most $1/|S| \le \hat{w}^{(j)}(\Gamma_S(i))/3$. Therefore, the total $\hat{w}^{(j)}$ LP weight of the bucketed sets S_s^{ji} for s such that $2^{-s} \ge 1/|S|^2$ is at least $\frac{2}{3}\hat{w}^{(j)}(\Gamma_S(i))$, and at least one of these bucketed sets has LP weight at least a $1/(2\log|S|)$-fraction of this, or at least $\hat{w}^{(j)}(\Gamma_S(i))/(3\log|S|) \ge 1/(3\log|S|)$. This gives a lower bound on the LP weight of the bucket which defines β_{ji}. Also, the heaviest bucket cannot be S_s^{ji} for s such that $2^{-s} \le 1/|S|^2$, since even the total weight of these buckets is at most $1/|S| = o(1/(3\log|S|))$. In particular, this means that $\beta_{ji} \ge 1/|S|^2$. Moreover, for s such that $2^{-s} = \beta_{ji}$, since the total conditional LP weight of S_s^{ji} is at least $1/(3\log|S|)$, and every vertex in the set has conditional LP value at most $2\beta_{ji}$, the cardinality of the set must be at least $1/(6\beta_{ji}\log|S|)$.

Now let us examine the second stage of bucketing. Note that for every $h \in \Gamma_S(i)$, we have $w_h \ge X_h^{(j)} = \hat{w}_h^{(j)}\cdot x_j \ge \beta_{ji}x_0 \ge 1/(|S|^2 m)$ (and, of course, $w_h \le 1$). Therefore, the number of non-empty buckets $\hat{S}_t^{ji}$ is at most $\log(|S|^2 m)$, and at least one of them must have cardinality at least $|S_s^{ji}|/\log(|S|^2 m)$, which, along with our lower bound on $|S_s^{ji}|$ above, gives us the required lower bound on $|\hat{\Gamma}_j(i)|$. ◀

Note that from the above proof, we also get upper-bounds on the number of values of β_{ji} and γ_{ji} that can produce non-empty buckets. In particular, we get the following bound:

▶ **Observation 25.** *The total number of possible values for β_{ji} is at most $2\log|S|$, and the total number of possible values for γ_{ji} is at most $\log(|S|^2 m)$. Along with the range of values for D, the total number of triples $\langle\beta,\gamma,D\rangle$ for which $J^D_{\beta,\gamma}$ is non-empty is at most $2\log^2|S|\log(|S|^2 m)$.*

The algorithm proceeds by separating the buckets corresponding to parameters for which the simple rounding (which samples a random subset of J of size $\tilde{\Omega}(J)$) makes progress towards an approximation guarantee of $\tilde{O}(A)$. If a large fraction of vertices in J_0 participate exclusively in such buckets, then the algorithm applies this rounding. The following lemma gives the analysis of the algorithm in this case.

▶ **Lemma 26.** *In Algorithm 2, if $|J_1| < |J_0|/2$, then with high probability the algorithm samples a subset $J_{\mathrm{ALG}} \subseteq J$ which covers an $\tilde{\Omega}(1)$-fraction of blue vertices, and a subset of S of size $\tilde{O}(A\cdot\mathrm{OPT})$ which covers all the red neighbors of J_{ALG}.*

Finally, we turn to the remaining case in Algorithm 2, when $|J_1| \le |J_0|/2$. The analysis of this case rests on a back-degree argument similar (though significantly more involved) to the argument in Lemma 10 for Red-Blue Set Cover. Indeed, we show the following:

▶ **Lemma 27.** *If $|J_1| \ge |J_0|/2$, then for β,γ,D,h_0 and the set J_{ALG} as defined by the algorithm in this case, we have $\sum_{j\in J_{\mathrm{ALG}}}\hat{x}_j^{(h_0)} \ge \frac{|J_0|Dx_0\beta}{\mathrm{OPT}}\cdot\frac{1}{4\log^2|S|\log(|S|^2m)\log m}$.*

Furthermore, for every vertex $j\in\tilde{J}$ (as defined by the algorithm), we have $\hat{x}_j^{h_0}\in[x_0\beta/(2\gamma), 4x_0\beta/\gamma]$.

We can now show that in this case, the algorithm makes progress towards an $\tilde{O}(m/A^2)$-approximation. Trading this off with the progress towards an $\tilde{O}(A)$-approximation as guaranteed by Lemma 26, we get an $\tilde{O}(m^{1/3})$-approximation by setting $A = m^{1/3}$.

▶ **Lemma 28.** *In Algorithm 2, if $|J_1| \ge |J_0|/2$, then with high probability, the algorithm makes progress towards an approximation guarantee of $\tilde{O}(m/A^2)$.*

Proof. Let us first bound the number of blue vertices covered by J_{ALG}. By Lemma 27, we have

$$\begin{aligned}
\hat{x}^{h_0}(J_{\mathrm{ALG}}) &\ge \frac{|J_0|x_0D\beta}{\mathrm{OPT}}\cdot\frac{1}{4\log^2|S|\log(|S|^2m)\log m} \\
&\ge \frac{x(J_0)D\beta}{\mathrm{OPT}}\cdot\frac{1}{4\log^2|S|\log(|S|^2m)\log m} && \text{since } \forall j\in J_0: 2x_0\ge x_j \\
&\ge \frac{x(J)D\beta}{\mathrm{OPT}}\cdot\frac{1}{4\log^2|S|\log(|S|^2m)\log m} && \text{by Lemma 23} \\
&> x(J)\cdot\frac{A}{x(J_0)}\cdot\frac{1}{4\log^2|S|\log(|S|^2m)\log m}. && \text{since } \forall\langle\beta,\gamma,D\rangle\in P_1: \frac{\beta D}{\mathrm{OPT}} > \frac{A}{x(J_0)}
\end{aligned}$$

Thus, since the conditioned LP solution satisfies the basic LP, we can apply Claim 21 to this solution and get that the size of the blue neighborhood of J_{ALG} can be bounded by

$$\begin{aligned}
|\Gamma_B(J_{\mathrm{ALG}})| &\ge \frac{1}{4e\ln(2k)\log k\log m}\cdot\frac{A}{x(J_0)}\cdot\frac{1}{4\log^2|S|\log(|S|^2m)\log m}\cdot|B| \\
&= \frac{1}{16\ln(2k)\log k\log^2|S|\log(|S|^2m)\log^2 m}\cdot\frac{A}{x(J_0)}\cdot|B|.
\end{aligned}$$

Note that by the LP constraints and Lemma 27, for every red neighbor $i \in \Gamma_R(J_{\text{ALG}})$, we have $\hat{y}_i^{h_0} \geq x_i^{h_0} \geq x_0\beta/(2\gamma)$, and so by (16), the rescaled solution $(\hat{w}_h^{h_0} \cdot 2\gamma/(x_0\beta))_{h\in S}$ is a fractional set cover for $\Gamma_R(J_{\text{ALG}})$. Thus, sampling every $h \in S$ with probability $\min\{1, 2\ln n \cdot \hat{w}_h^{h_0} \cdot 2\gamma/(x_0\beta)\}$ produces a valid set cover with high probability. It remains to analyze the size of this set cover. Indeed, since $\hat{w}^{h_0}(S) \leq \text{OPT}$, our sampling procedure produces a set of expected size

$$\begin{aligned}
\mathbb{E}[|S_{\text{ALG}}|] &\leq \frac{4\gamma \ln n}{x_0\beta} \cdot \text{OPT} \\
&\leq 8\ln n \cdot \frac{\gamma}{\beta} \cdot \frac{|J_0|}{x(J_0)} \cdot \text{OPT} && \text{since } \forall j \in J_0 : x_j \leq 2x_0 \\
&\leq 8\ln n \cdot \frac{\gamma}{\beta} \cdot \frac{m}{x(J_0)} \cdot \text{OPT} \\
&< 8\ln n \cdot \frac{1}{A} \cdot \frac{m}{x(J_0)} \cdot \text{OPT}, && \text{since } \forall \langle \beta, \gamma, D\rangle \in P_1 : \frac{\beta}{\gamma} > A
\end{aligned}$$

and so with high probability we have $|S_{\text{ALG}}| = O(\text{OPT} \cdot m \log n/(A \cdot x(J_0)))$.

Putting our two bounds together, we get that in this case, the algorithm makes progress towards an approximation guarantee of

$$\frac{|B|}{|\Gamma_B(J_{\text{ALG}})|} \cdot \frac{|S_{\text{ALG}}|}{\text{OPT}} = \tilde{O}(1) \cdot \frac{x(J_0)}{A} \cdot \frac{m}{A \cdot x(J_0)} = \tilde{O}(1) \cdot \frac{m}{A^2}. \qquad \blacktriangleleft$$

References

1 V. P. Abidha and Pradeesha Ashok. Red blue set cover problem on axis-parallel hyperplanes and other objects. *CoRR*, abs/2209.06661, 2022. doi:10.48550/arXiv.2209.06661.

2 Michael Alekhnovich, Samuel R. Buss, Shlomo Moran, and Toniann Pitassi. Minimum propositional proof length is np-hard to linearly approximate. *J. Symb. Log.*, 66(1):171–191, 2001. doi:10.2307/2694916.

3 Pradeesha Ashok, Sudeshna Kolay, and Saket Saurabh. Multivariate complexity analysis of geometric red blue set cover. *Algorithmica*, 79(3):667–697, 2017. doi:10.1007/s00453-016-0216-x.

4 Pranjal Awasthi, Avrim Blum, and Or Sheffet. Improved guarantees for agnostic learning of disjunctions. In Adam Tauman Kalai and Mehryar Mohri, editors, *COLT 2010 - The 23rd Conference on Learning Theory, Haifa, Israel, June 27-29, 2010*, pages 359–367. Omnipress, 2010. URL: http://colt2010.haifa.il.ibm.com/papers/COLT2010proceedings.pdf#page=367.

5 Robert D. Carr, Srinivas Doddi, Goran Konjevod, and Madhav V. Marathe. On the red-blue set cover problem. In David B. Shmoys, editor, *Proceedings of the Eleventh Annual ACM-SIAM Symposium on Discrete Algorithms, January 9-11, 2000, San Francisco, CA, USA*, pages 345–353. ACM/SIAM, 2000. URL: http://dl.acm.org/citation.cfm?id=338219.338271.

6 Timothy M. Chan and Nan Hu. Geometric red-blue set cover for unit squares and related problems. *Comput. Geom.*, 48(5):380–385, 2015. doi:10.1016/j.comgeo.2014.12.005.

7 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, September 7-9, 2016, Paris, France*, volume 60 of *LIPIcs*, pages 4:1–4:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. doi:10.4230/LIPIcs.APPROX-RANDOM.2016.4.

8 Eden Chlamtác, Michael Dinitz, Christian Konrad, Guy Kortsarz, and George Rabanca. The densest k-subhypergraph problem. *SIAM J. Discret. Math.*, 32(2):1458–1477, 2018. doi:10.1137/16M1096402.

9 Eden Chlamtác, Michael Dinitz, and Yury Makarychev. Minimizing the union: Tight approximations for small set bipartite vertex expansion. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19*, pages 881–899. SIAM, 2017. doi:10.1137/1.9781611974782.56.

10 Irit Dinur and Shmuel Safra. On the hardness of approximating label-cover. *Inf. Process. Lett.*, 89(5):247–254, 2004. doi:10.1016/j.ipl.2003.11.007.

11 Michael Elkin and David Peleg. The hardness of approximating spanner problems. *Theory Comput. Syst.*, 41(4):691–729, 2007. doi:10.1007/s00224-006-1266-2.

12 Michael H. Goldwasser and Rajeev Motwani. Intractability of assembly sequencing: Unit disks in the plane. In Frank K. H. A. Dehne, Andrew Rau-Chaplin, Jörg-Rüdiger Sack, and Roberto Tamassia, editors, *Algorithms and Data Structures, 5th International Workshop, WADS '97, Halifax, Nova Scotia, Canada, August 6-8, 1997, Proceedings*, volume 1272 of *Lecture Notes in Computer Science*, pages 307–320. Springer, 1997. doi:10.1007/3-540-63307-3_70.

13 Raghunath Reddy Madireddy and Apurva Mudgal. A constant-factor approximation algorithm for red-blue set cover with unit disks. *Algorithmica*, 85(1):100–132, 2023. doi:10.1007/s00453-022-01012-z.

14 Raghunath Reddy Madireddy, Subhas C. Nandy, and Supantha Pandit. On the geometric red-blue set cover problem. In Ryuhei Uehara, Seok-Hee Hong, and Subhas C. Nandy, editors, *WALCOM: Algorithms and Computation - 15th International Conference and Workshops, WALCOM 2021, Yangon, Myanmar, February 28 - March 2, 2021, Proceedings*, volume 12635 of *Lecture Notes in Computer Science*, pages 129–141. Springer, 2021. doi:10.1007/978-3-030-68211-8_11.

15 Pauli Miettinen. On the positive-negative partial set cover problem. *Inf. Process. Lett.*, 108(4):219–221, 2008. doi:10.1016/j.ipl.2008.05.007.

A Adapting and Applying our Algorithm to Partial Red-Blue Set Cover

Let us now consider the variation in which we are given a parameter $\hat{k}$, and are only required to cover at least $\hat{k}$ elements in a feasible solution. The algorithm and analysis work with almost no change other than the following.

In the algorithm, the stopping condition of the loop is of course no longer once we have covered all blue elements, but once we have covered at least $\hat{k}$ of them.

A slightly more subtle change involves the analysis of the LP rounding in the final iteration. The notion of progress towards a certain approximation guarantee may not be valid if the ratio of red elements to blue elements covered is still as small as required, but the number of blue elements added is far more than we need. Rather than derandomize the rounding, one can show that it succeeds (despite this issue) with high probability. Let us briefly sketch the argument here.

First, note that we can always preemptively discard any sets with more than OPT red elements, and so we may assume that $r_\alpha \leq \mathrm{OPT}$. Suppose we need to cover an additional k^* elements in order to reach the target of $\hat{k}$ blue elements total. Since our bound on $\mathbb{E}[|\Gamma_{R_\alpha}(J^*)|]$ is a linear function of our bound on $\mathbb{E}[|J^* \setminus J_+|]$, by a Chernoff bound we have $|\Gamma_{R_\alpha}(J^*)| = \tilde{O}(r_\alpha A)$ with all but exponentially small probability. On the other hand, $|\Gamma_B(J^*)|$ is always at most $|B|$, so by Markov, we have

$$\mathrm{Prob}\left[|\Gamma_B(J^*)| \leq \frac{\mathbb{E}[|\Gamma_B(J^*)|]}{2}\right] \leq \frac{|B| - \mathbb{E}[|\Gamma_B(J^*)|]}{|B| - \mathbb{E}[|\Gamma_B(J^*)|]/2} \leq 1 - \frac{1}{2|B|}.$$

Thus, repeating the rounding a polynomial number of times (in a given iteration), with all but exponentially small probability we can find a set $\hat{J} \subseteq J$ that satisfies both

$$|\Gamma_B(\hat{J})| \geq \frac{\mathbb{E}[|\Gamma_B(J^*)|]}{2} \qquad \text{and} \qquad |\Gamma_{R_\alpha}(\hat{J})| = \tilde{O}(r_\alpha A).$$

Now if $\mathbb{E}[|\Gamma_B(J^*)|] \leq 2k^*$, then we have the required ratio and bound on the number of new red elements by the previous analysis. If $\mathbb{E}[|\Gamma_B(J^*)|] > 2k^*$, then this will be the last iteration, as we will cover at least the required k^* additional blue elements, and the number of red elements added at this final stage is at most $\tilde{O}(r_\alpha A) \leq \tilde{O}(A \cdot \mathrm{OPT})$, so we maintain the desired approximation ratio.

B Additional LP Constraints for MMSA$_4$

The following is a complete list of lifted constraints that we use in addition to the basic LP relaxation for MMSA$_4$:

$$\begin{aligned}
&X_j^{(h)} = X_h^{(j)} && \forall j \in J, \forall h \in S \\
&X_j^{(j)} = x_j && \forall j \in J \\
&\sum_{h \in \Gamma_S(i)} X_h^{(j)} \geq X_i^{(j)} && \forall j \in J \forall i \in R \\
&X_j^{(j)} \leq X_i^{(j)} && \forall j \in J \forall i \in \Gamma_R(j) \\
&0 \leq X_a^{(j)} \leq x_j && \forall j \in J \forall a \in \{j\} \cup R \cup S \\
&\sum_{h' \in S} w_{h'}^{(h)} \leq \mathrm{OPT} && \forall h \in S \\
&\sum_{\ell \in B} X_\ell^{(h)} \geq |B|/(\log k \log m) w_h && \forall h \in S \\
&X_\ell^{(h)} \leq \sum_{j \in \Gamma_J(\ell)} X_{\ell,j}^{(h)} \leq 2e \ln(2k) X_\ell^{(h)} && \forall h \in S \forall \ell \in B \\
&\Delta X_j^{(h)} \leq \sum_{\ell \in \Gamma_B(j)} X_{\ell,j}^{(h)} \leq 2\Delta X_j^{(h)} && \forall h \in S \forall j \in J \\
&0 \leq X_{\ell,j}^{(h)} \leq X_j^{(h)}, X_\ell^{(h)} \leq w_h && \forall h \in S \forall \ell \in B \forall j \in J \\
&\sum_{h' \in \Gamma_S(i)} X_{h'}^{(h)} \geq X_i^{(h)} && \forall h \in S \forall i \in R \\
&X_j^{(h)} \leq X_i^{(h)} && \forall h \in S \forall (j,i) \in E(J,R) \\
&0 \leq X_a^{(h)} \leq w_h && \forall h \in S \forall a \in B \cup J \cup R \cup S
\end{aligned}$$

Algorithm 2 Approximation Algorithm for MMSA$_4$.

Input: B, J, R, S, E
Guess OPT, Δ and solve the LP ▷ e.g. using binary search
Choose parameter s such that the LP weight of the bucket $J_s = \{j \in J \mid 2^{-s} \le x_j \le 2^{-(s-1)}\}$, that is, $\sum_{j\in J_s} x_j$ is maximized, and let $x_0 = 2^{-s}$ and $J_0 = J_s$.
for every $j \in J_0$ and $i \in \Gamma_R(j)$ **do**
  Choose a new parameter s such that the conditioned LP weight of the bucket $S^{ji}_s = \{h \in S \mid 2^{-s} \le \hat{w}^{(j)}_h \le 2^{-(s-1)}\}$, that is, $\sum_{h\in S^{ji}_s} \hat{w}^{(j)}_h$, is maximized, and let $\beta_{ji} = 2^{-s}$.
  Choose parameter t such that the sub-bucket $\hat{S}^{ji}_t = \{h \in S^{ji}_s \mid 2^{-t} \le w_h \le 2^{-(t-1)}\}$ has maximum cardinality $|\hat{S}^{ji}_t|$, and let $\gamma_{ji} = 2^{-t}$ and $\hat{\Gamma}_j(i) = \hat{S}^{ji}_t$.
end for
for every $j \in J_0$ and β, γ **do**
  Let $\Gamma^R_{\beta,\gamma}(j) = \{i \in \Gamma_R(j) \mid \beta_{ji} = \beta, \gamma_{ji} = \gamma\}$.
  Let $\Gamma^S_{\beta,\gamma}(j) = \bigcup_{i\in\Gamma^R_{\beta,\gamma}(j)} \hat{\Gamma}_j(i)$.
end for
for every β, γ and $D \in \{2^{s-1} \mid s \in \lceil \log |S| \rceil\}$ **do**
  Let $J^D_{\beta,\gamma} = \{j \in J_0 \mid \Gamma^R_{\beta,\gamma}(j) \ne \emptyset, |\Gamma^S_{\beta,\gamma}(j)| \in [D, 2D]\}$.
  Let $T^D_{\beta,\gamma} = \{\langle j, \Gamma^R_{\beta,\gamma}(j), \Gamma^S_{\beta,\gamma}(j)\rangle \mid j \in J^D_{\beta,\gamma}(j)\}$.
end for
Let $P_1 = \{\langle \beta, \gamma, D \mid \beta/\gamma > A, \beta D > A \cdot \mathrm{OPT}/x(J_0)\rangle\}$, and $J_1 = \bigcup_{\langle\beta,\gamma,D\rangle\in P_1} J^D_{\beta,\gamma}$.
if $|J_1| < |J_0|/2$ **then**
  Let $J_{\mathrm{ALG}} = \emptyset$.
  for all $j \in J_0 \setminus J_1$ **do**
    Independently add j to J_{ALG} with probability x_0.
  end for
  Let $S_{\mathrm{ALG}} = \emptyset$.
  for every β **do**
    Let $S_\beta = \bigcup_{j\in J_{\mathrm{ALG}}} \bigcup_\gamma \Gamma^S_{\beta,\gamma}(j)$.
    for all $h \in S_\beta$ **do**
      Independently add h to S_{ALG} with probability $\min\{1, \beta \cdot 12 \log |S| \log(|S|^2 m) \ln n\}$.
    end for
  end for
else if $|J_1| \ge |J_0|/2$ **then**
  Choose $\langle \beta, \gamma, D\rangle \in P_1$ that maximize the cardinality $|J^D_{\beta,\gamma}|$, and let $J_2 = J^D_{\beta,\gamma}$.
  Let $S_{\tilde{D}} = \{h \in S \mid \{j' \in J_2 \mid h \in \Gamma^S_{\beta,\gamma}(j')\}| \in [\tilde{D}, 2\tilde{D}]\}$ for every $\tilde{D} \in \{2^{s-1} \mid s \in \lceil \log |J_2| \rceil\}$.
  Choose $\tilde{D}$ that maximizes the cardinality $|\{\langle j, h\rangle \in J_2 \times S_{\tilde{D}} \mid h \in \Gamma^S_{\beta,\gamma}(j)\}|$, and let $\tilde{S} = S_{\tilde{D}}$.
  Choose $h_0 \in \tilde{S}$ that maximizes the total LP value $\sum_{j\in J_2:\Gamma^S_{\beta,\gamma}(j)\ni h_0} \hat{x}^{(h_0)}_j$.
  Let $J_{\mathrm{ALG}} = \{j \in J_2 \mid h_0 \in \Gamma^S_{\beta,\gamma}(j)\}$.
  Let $S_{\mathrm{ALG}} = \emptyset$.
  for every $h \in \bigcup_{j\in J_{\mathrm{ALG}}} \Gamma_S(\Gamma_R(j))$ **do**
    Independently add h to S_{ALG} with probability $\min\{1, \hat{w}^{h_0}_h \cdot 4\gamma \ln n/(x_0\beta)\}$.
  end for
end if