Electronic Colloquium on Computational Complexity, Report No. 38 (2003)

Asymmetric k-center is $\log^* n$-hard to Approximate

Julia Chuzhoy* Sudipto Guha† Sanjeev Khanna‡ Joseph (Seffi) Naor§

May 15, 2003

Abstract

We show that the asymmetric k-center problem is $\Omega(\log^* n)$-hard to approximate unless NP $\subseteq$ DTIME$(n^{poly(\log\log n)})$. Since an $O(\log^* n)$-approximation algorithm is known for this problem, this essentially resolves the approximability of this problem. This is the first natural problem whose approximability threshold does not polynomially relate to the known approximation classes. Our techniques also resolve the approximability threshold of the weighted metric k-center problem. We show that it is hard to approximate to within a factor of $3-\epsilon$ for any $\epsilon > 0$.

1 Introduction

The input to the asymmetric k-center problem consists of a set V of vertices of a complete digraph G, and a weight (or distance) function $c : V \times V \to \mathbb{R}^+ \cup \{0\}$. The weight function c must satisfy the (directed) triangle inequality, i.e., $\forall v \in V$, $c_{vv} = 0$ and $\forall u, v, w \in V$, $c_{uv} + c_{vw} \geq c_{uw}$. The goal is find a set S of k vertices (denoted as centers) and to assign all the vertices to these centers, such that the maximal distance of a vertex from its center is minimized. To state mathematically, we want to find a subset $S \subseteq V$ of size k, that minimizes

$$\max_{v\in V} \min_{u\in S} c_{uv}$$

If the function c is assumed to be symmetric as well, the above problem is the (metric) k-center problem. This is one of the early problems for which approximation algorithms were designed, and an optimal approximation ratio of 2 is known [9]. Subsequent to the solution of this problem a significant number of other problems in location theory were solved (see [11]); however the approximability threshold of the asymmetric case remained open[1].

*Department of Computer Science, Technion, Haifa, Israel. Email: cjulia@cs.technion.ac.il

†Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19103. Email:sudipto@cis.upenn.edu

‡Department of Computer and Information Science, University of Pennsylvania, Philadelphia, PA 19103. Email:sanjeev@cis.upenn.edu

§Department of Computer Science, Technion, Haifa, Israel. Email: naor@cs.technion.ac.il

[1]The problem is inapproximable if the triangle inequality does not hold.

ISSN 1433-8092

In a significant step, Panigrahy and Vishwanathan [10] gave an $O(\log^* n)$ approximation algorithm, which was subsequently improved by Archer [1] to $O(\log^* k)$. In this paper we show that these results are asymptotically tight, unless NP $\subseteq$ DTIME$(n^{poly(\log\log n)})$. This is a lower bound for a natural problem which does not conform to the known classes of approximation (see [2]). The result of [8] is already a step in this direction, which shows a poly-logarithmic inapproximability for the group Steiner problem. However, our paper demonstrates a problem whose approximation factor is not polynomially related to the known classes. We remark that our result holds even for weaker assumption. We show that for every constant i, asymmetric k-center is $\Omega(\log^* n)$-hard to approximate, unless $NP \subseteq DTIME(n^{\log^{(i)} n})$. We note here that Halperin, Kortsarz, and Krauthgamer [7] have independently obtained a similar result.

Our results build on the sequence of recent results leading to a $(k-1-\epsilon)$ hardness for k-hypergraph covering [6, 5, 3, 4]. The result of [4] can be viewed as a construction of an instance of Set Cover from an instance of Gap 3SAT(3) with a strong bi-criterion gap. If the formula is satisfiable then a $1/(k-1-\epsilon)$ fraction of the sets are sufficient to cover all the elements. If the formula is unsatisfiable then a fraction $1-\epsilon$ of the sets cover at most a $1-f(k,\epsilon)$ fraction of the elements

At a high level, our reduction works as follows: Given an instance φ of 3-SAT(3), we build a directed graph, with $h+1 = \Theta(log^*(N))$ layers of vertices (N is the number of vertices, $N = n^{poly(\log\log n)}$). For each pair of consecutive layers i, $(i+1)$, there are directed edges from some layer i vertices to some layer $(i+1)$ vertices. This graph is translated into an instance of k-center as follows. The set of vertices remains the same. For some i, consider layer i vertex v and layer $(i+1)$ vertex u. The distance $c(u,v) = \infty$. If there is an edge (v,u) in the original graph, then $c(v,u) = 1$. Otherwise, $c(v,u) = \infty$. The distances between any other pair of vertices are implied by these distances.

Layer 0 of vertices consists of only one vertex, which is connected to every vertex in layer 1. For any two other consecutive layers $i, (i+1)$, we build a Set Cover instance, where layer i vertices serve as sets, and layer $(i+1)$ vertices as elements. There is a directed edge from layer i vertex v to layer $i+1$ vertex u iff the element corresponding to u belongs to the set corresponding to u.

If the formula φ is a yes-instance, we are able to cover all the vertices by $k+1$ centers with radius 1, i.e., apart from the vertex at level 0 (which we include in our solution), we will find solutions to all the Set Cover instances, using in total only k sets.

If φ is a no-instance, we will prove that it is impossible to cover all the vertices by $k+1$ centers with radius $(h-1)$. To do this, it is enough to show that it is impossible to choose k vertices in layer 1 that cover (with radius $(h-1)$) all the vertices in layer h, since we can assume that every solution uses only vertices in layers 0 and 1. Indeed, any solution must contain the layer 0 vertex (because it is impossible to cover this vertex otherwise), which covers all the vertices, except for layer h. As we are allowed to use radius $h-1$, there is no point in taking any vertex v of some layer $i > 1$ to the solution: the predecessor of v in layer 1 can cover (with radius $h-1$) all the vertices v can cover.

We remark that we could recast the proofs of the papers [3, 4] to construct the family of Set Cover instances we need. However, we will not be requiring the full machinery of the hypergraph constructions. Since we will be required to compute the various parameters of the Set Cover instance and how the layers interact, we will present the full proof with due apologies to the readers well familiar with hypergraph hardness constructions. In fact, our construction of the Set Cover instances is very similar to [3], and many proofs are in the same spirit as in [4].

Finally, we would like to remark that we did not try to optimize the constant in the $\Theta(\log^* n)$ hardness, but our goal was rather to show the simplest construction. In fact, using Set Cover construction similar to [4], results in a better constant, though more complicated construction. We defer the optimization of the constant to the journal version.

The rest of this paper is organized as follows. In Section 2 we will present the preliminaries of the Raz verifier. In section 3 we will present the Set Cover instances we will construct given a formula φ. In Section 4 we will complete the reduction to asymmetric k-center. The results will be used in Section 5 to show tight lower bounds for the weighted (metric) k-center problem. An explicit construction of the Integrality gap appears in Appendix C.

2 Preliminaries: The Raz Verifier for Gap 3SAT(3)

The 3SAT(3) problem is defined as follows. We are given a 3CNF(3) formula, with n variables and n clauses. Each clause has exactly 3 different variables and each variable appears in exactly 3 clauses.

An instance φ of the problem is defined to be an yes-instance, if there is a satisfying assignment. It is defined to be a no-instance, if every assignment to φ satisfies at most $(1-\epsilon)$ fraction of clauses simultaneously. The following is a well-know fact:

Theorem 1 *There is an ϵ, such that it is NP-hard to distinguish between the Yes and the No instances of 3SAT(3).*

We will use the Raz verifier for 3SAT(3). The verifier with ℓ repetitions works as follows. We randomly choose ℓ clauses, and in each clause we randomly choose one of its variables, called a **distinguished variable**. Then we send the indices of the clauses to prover 1 and the indices of the distinguished variables to prover 2. The answer of prover 1 is a 3ℓ-bit string specifying the assignments to all the variables that appeared in the clauses, which satisfy each one of these clauses. The answer of prover 2 is an ℓ-bit string specifying the assignments to all the distinguished variables.

The verifier performs the following test: for every clause C that was randomly chosen previously, we check that the assignment to its variables, returned by prover 1, satisfies the clause. Also, we check that the distinguished variable of C gets identical assignment in the answers of the two provers. We do not perform any additional tests. For example, it is possible for the same variable to appear in two different clauses and to get different assignments: we do not check that.

This proof system can be viewed as follows: There is a set Y of variables, where each variable $y \in Y$ corresponds to a query (with ℓ repetitions) to prover 1. Similarly, there is a set Z of variables corresponding to all the possible queries to prover 2. An assignment to y-variable is a 3ℓ-bit string that satisfies all the corresponding clauses, and an assignment to z-variable is an ℓ-bit string. Let r be some random string, and let $y \in Y$, $z \in Z$ be the variables corresponding to the two queries that are asked, given r. We say that there is a constraint $C_r(y, z)$. The constraint is: the assignments to variables y, z must be consistent. Thus, there is a one-to-one correspondence between the random strings r and the constraints.

Theorem 2 *There is a constant $0 < \alpha < 1$, such that:*

- *For any yes-instance φ, there is an assignment to the variables in Y, Z, such that all the constraints are satisfied.*
- *For any no-instance φ, no assignment can satisfy a fraction $\geq 2^{-\alpha\ell}$ of constraints.*

Notation: We denote the set of all the possible random strings by R. Clearly, $|R| = (3n)^\ell$. The set of all the possible queries for provers 1 and 2 are denoted by Q_1 and Q_2. We denote $Q = |Q_1| = |Q_2| = |Y| = |Z| = n^\ell$. Note that for each possible query for prover 1 (prover 2), there are exactly 3^ℓ possible queries for prover 2 (prover 1). We denote the set of all the possible assignments to variables Y and Z by A_Y and A_Z, $|A_Y| = 8^\ell$, $|A_Z| = 2^\ell$, and denote $A = 8^\ell$, so $|A_Y|, |A_Z| \leq A$.

Consider variables $y \in Y$, $z \in Z$, such that there is a constraint $C_r(y, z)$. For every assignment to y, there is at most one assignment to z that will satisfy the constraint.

3 The Basic Set Cover Instance

Given an instance φ of 3-Sat(3), we build the basic Set Cover instance, based on the Raz verifier. Our final construction of k-center instance uses several such basic Set Cover instances, connected back-to-back, while different basic Set Cover instances have different parameters.

There are two parameters that determine a basic Set Cover instance:

- $d \geq \frac{64}{\alpha}$ is the number of sets in which each element participates. If φ is a yes-instance, then there is a set cover that uses a fraction $\frac{1}{d-2}$ of sets.
- b - the size of a basic set block - to be described later.

The Set Cover constructed will correspond to the Raz verifier with $\ell = d^7$.

3.1 The Sets

The sets in our Set Cover instance are divided into variable blocks of sets, each block corresponding to a variable in the Raz verifier. Each variable block of sets is further subdivided into assignment blocks of sets, which in turn consist of basic set blocks.

Variable blocks: For each variable $x \in Y \cup Z$ of Raz verifier, there is a **variable block** $B(x)$ of sets. The number of such blocks is the total number of variables in Raz verifier, which is $2n^\ell$.

Assignment blocks: Variable blocks of sets are further divided into **assignment blocks** of sets. Consider a variable $x \in Y \cup Z$, and let A_x be the set of all the possible assignments to x. For each subset $\mathcal{A} \subseteq A_x$, there is an assignment block $B(x, \mathcal{A})$.

Let $p = 1 - \frac{1}{d-2}$. For each subset of assignments $\mathcal{A} \subseteq A_x$, we define the **normalized weight** $w(\mathcal{A})$ of $\mathcal{A}$ to be the probability of choosing $\mathcal{A}$, when performing the following trial. Each assignment $a \in A_x$ is chosen independently with probability p. Thus, $w(\mathcal{A}) = p^{|\mathcal{A}|}(1-p)^{|A_x \setminus \mathcal{A}|}$. We need the weight of $\mathcal{A}$ to be integral, thus we define the (non-normalized) weight of $\mathcal{A}$ to be

$$W(\mathcal{A}) = (d-2)^A w(\mathcal{A})$$

Basic set blocks: Each assignment block $B(x, \mathcal{A})$ is further subdivided into $W(\mathcal{A})$ **basic set blocks**. Each basic set block consists of b sets[2]. To conclude, here is how a variable block of sets looks like.

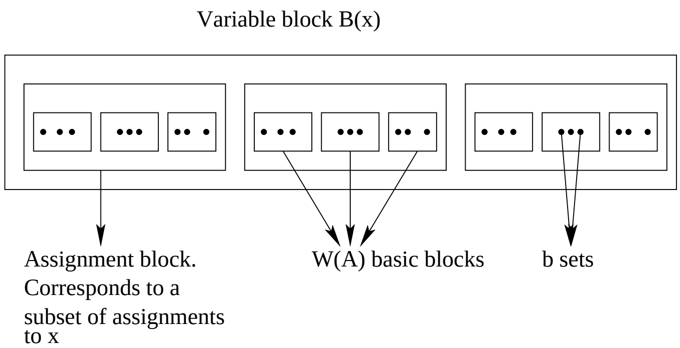

Figure 1: Variable block $B(x)$

In each variable block, the total normalized weight of the assignment blocks is 1. Therefore, the number of basic set blocks belonging to each variable is at most $d^A = d^{8^l}$.

3.2 The Elements

The elements are divided into constraint blocks, corresponding to the constraints in Raz verifier, and each constraint block is further subdivided into basic element blocks.

Constraint blocks: For each constraint $C_r(y, z)$, $y \in Y$, $z \in Z$ in Raz verifier, there is a constraint block of elements $B(C)$. Thus, the number of constraint blocks is $(3n)^\ell$.

Basic element blocks: Each constraint block is further subdivided into basic element blocks. Consider the constraint $C_r(y, z)$, $y \in Y$, $z \in Z$. Let $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1} \subseteq A_Y$ be subsets of assignments

[2]Note: given the assignment block $B(x, \mathcal{A})$, the corresponding set blocks and their sets **do not** represent the assignments $a \in \mathcal{A}$. In fact, they do not have any meaning, except that the number of the set blocks is $W(\mathcal{A})$. Also, the weights of the blocks do not have any meaning in the final Set Cover construction. It is convenient to use them in the analysis, and they also correspond to the number of basic blocks in each assignment block.

to y, and $\mathcal{B} \subseteq A_Z$ be a subset of assignments to z. Denote $\mathcal{A} = \mathcal{A}_1 \cap \cdots \cap \mathcal{A}_{d-1}$. Then there is a basic element block $B_{y,z}(\mathcal{A}_1, \ldots, \mathcal{A}_{d-1}, \mathcal{B})$, iff there is no pair of assignments $a_1 \in \mathcal{A}$, $a_2 \in \mathcal{B}$ that are consistent (i.e., a_1 and a_2 satisfy the constraint $C_r(y, z)$).

The number of basic element blocks in each constraint block is at most $2^{d \cdot 8^\ell}$.

Suppose there is a basic element block $B_{y,z}(\mathcal{A}_1, \ldots, \mathcal{A}_{d-1}, \mathcal{B})$. Then for each sequence of sets $v_1, \ldots, v_{d-1}, u$, such that v_i (for $1 \leq i \leq d-1$) belongs to the assignment block $B(y, \mathcal{A}_i)$, and u belongs to the assignment block $B(z, \mathcal{B})$, there is an element e in the basic element block that belongs to these sets and only to them.

The number of elements in each basic element block is at most: $\left(b \cdot d^{8^\ell}\right)^d = b^d \cdot d^{d 8^\ell}$.

3.3 Properties of the Basic Set Cover instance

Definition: Given a (partial) solution to the Set Cover instance, we say that a block of elements is **covered** iff **all** its elements are covered.

Definition: Given a block of sets, we say that this block is **chosen** iff **all** the sets in this block are in the solution.

Proposition 1 *Consider any solution to a basic Set Cover instance. A basic block is covered iff one of its corresponding assignment blocks is chosen.*

Proof: Suppose there is an element block $B_{y,z}(\mathcal{A}_1, \ldots, \mathcal{A}_{d-1}, \mathcal{B})$, such that none of the assignment blocks $B(y, \mathcal{A}_1), \ldots, B(y, \mathcal{A}_{d-1}), B(z, \mathcal{B})$ are chosen. Then there are sets $v_1, \ldots, v_{d-1}, u$, such that for all $i : 1 \leq i \leq d-1$, $v_i \in B(y, \mathcal{A}_i)$, and $u \in B(z, \mathcal{B})$, which are not chosen. But then there is an element in the element block, that belongs to the sets $v_1, \ldots, v_{d-1}, u$ and only to them. Thus, this element is not covered. □

We can now prove the following:

Theorem 3 (Yes Instance) *If the original 3SAT(3) formula φ is a yes-instance, then it is possible to cover all the elements by using a fraction $(1-p) = \frac{1}{d-2}$ of sets.*

Proof: Suppose φ is a yes-instance, then there is an assignment A that satisfies all the variables. For each variable x, take into the solution all the sets in the assignment blocks $B(x, \mathcal{A})$, such that the assignment set $\mathcal{A}$ does not contain the assignment $A(x)$.

To see that we use only a fraction $(1-p)$ of sets: consider some variable $x \in Y \cup Z$ and its assignment $A(x)$. The total normalized weight of all the subsets of assignments to x that do not contain $A(x)$ is exactly $(1-p) = \frac{1}{d-2}$ (since the probability to choose a subset of assignments that does not contain $A(x)$ is exactly $(1-p)$).

Now suppose there is some constraint block $B(C_r(y, z))$ which is not covered, $y \in Y$, $z \in Z$. Then there is a basic set of elements, corresponding to subsets of assignments $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1}, \mathcal{B}$, where $\mathcal{A}_i \subseteq A_Y$ for $1 \leq i \leq d-1$, $\mathcal{B} \subseteq A_Z$. Since the basic element set is not covered, all the corresponding assignment blocks are not chosen. Therefore, all the subsets of assignments $\mathcal{A}_i$, $1 \leq i \leq d-1$ contain $A(x)$ and $\mathcal{B}$ contains $A(y)$. But $A(x)$ and $A(y)$ are consistent, and thus, by the definition of basic element blocks, such a basic element block does not exist. □

Theorem 4 (No-instance) *Suppose φ is a no-instance, and we have a solution, where a fraction $\delta < \frac{1}{10}$ of the basic set blocks are chosen. Then the the fraction of covered constraint blocks is at most $8\delta + \frac{1}{d}$.*

To prove the theorem we will first start with the following definition:

Definition: Given a variable x, let $I(x)$ be the family of all the subsets of assignments $\mathcal{A}$ to x, such that the assignment block $B(x, \mathcal{A})$ is **not chosen**. Define a weight of variable x to be the sum of the normalized weights of all the assignment subsets in $I(x)$.

Note that if $2V$ is the total number of variables, and at most a fraction δ of the basic set blocks are chosen, then the total weight of all the variables is $\geq (1-\delta) \cdot 2V$. We will use the following claim.

Claim 1 *Let $x \in X$ be some variable, and suppose the weight of x is $\geq \frac{1}{4}$. Then there are $(d-1)$ subsets of assignments $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1} \in I(x)$ whose intersection size is less than d^5.*

Proof: Suppose this lemma is not true. Consider the family $I(x)$ of subsets of assignments. Its normalized weight is $\geq \frac{1}{4}$. In Appendix, section A, we show that such a family must have $(d-1)$ subsets whose intersection size is less than d^5. □

Denote by X' the set of variables whose weight is $\geq \frac{1}{4}$. Let $Y' = X' \cap Y$, and $Z' = X' \cap Z$. Let C' be the subset of all the constraints between variables in Y' and in Z'. We start with a simple claim.

Claim 2 *If the fraction of basic set blocks chosen is at most δ, then the number of constraints in C' is at least a fraction $(1-8\delta)$ of all the constraints.*

Proof: First observe that $|X'| \geq (1-2\delta) \cdot 2V$. Suppose this is not true. Then the total weight of variables in X' is $< (1-2\delta) \cdot 2V$, and the total weight of all the other variables is $< 4\delta V \cdot \frac{1}{4}$. Thus, the total weight of all the variables is less than $(1-\delta) \cdot 2V$, which is not true.

Therefore, $|Y \setminus Y'| \leq 4\delta V$, and $|Z \setminus Z'| \leq 4\delta V$. Since each variable participates in the same number of constraints, there is a fraction $\leq 4\delta$ of constraints in which variables in $Y \setminus Y'$ participate, and a fraction $\leq 4\delta$ of constraints in which variables in $Z \setminus Z'$ participate. Thus, there is a fraction $\geq (1-8\delta)$ of constraints $C_r(y, z)$, where $y \in Y'$ and $z \in Z'$. □

Consider the set of constraints C' between the variables in Y' and Z'. Suppose we showed that for at least a fraction $(1-\frac{1}{d})$ of constraints in C', the constraint blocks are not covered. Then a fraction $\geq (1-8\delta)(1-\frac{1}{d}) \geq (1-8\delta-\frac{1}{d})$ of the total constraint blocks are not covered, proving Theorem 4.

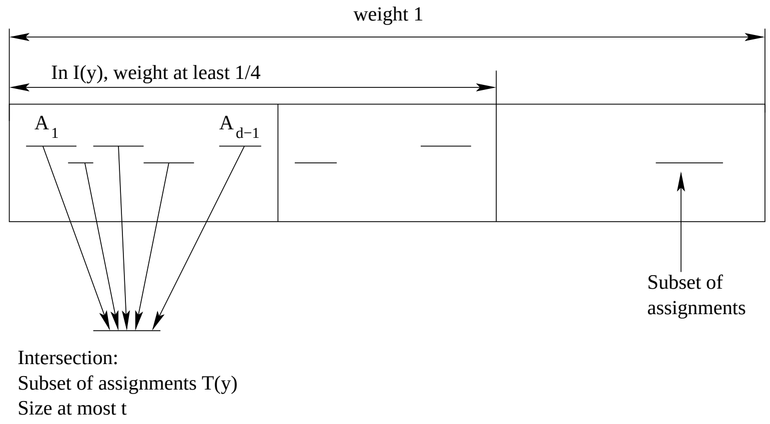

Figure 2: Variable $y \in Y'$

Consider some variable $y \in Y'$. The total normalized weight of assignments in $I(y)$ is at least $\frac{1}{4}$ (since $y \in Y'$). By Claim 1, there are $(d-1)$ assignment subsets $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1} \in I(y)$, whose intersection size is less than $t = d^5$. We denote this intersection by $T(y)$.

If Theorem 4 is not true, we will show that a formula φ which is a no-instance will have more than a fraction $\geq 2^{-\alpha\ell}$ of satisfied constraints. Consider some $y \in Y'$. The assignment to y is chosen uniformly at random from $T(y)$.

Consider some $z \in Z'$. We denote by $P(z)$ the set of variables $y \in Y'$, such that the constraint $C(y,z)$ exists and is covered. Given an assignment u to z, $count(u)$ is the number of variables $y \in P(z)$, such that u is consistent with at least one of the assignments in $T(y)$. We choose an assignment u with largest $count(u)$. The following Lemma (on similar analysis lines as in [4]) is proved in the Appendix B.

Lemma 1 *There is an assignment u to z, such that $count(u) \geq \frac{1}{2t(d-2)^t} \cdot |P(z)|$.*

Lemma 2 *If the formula φ is a no-instance, at least a fraction $(1 - \frac{1}{d})$ of constraint blocks corresponding to the constraints in C' are not covered.*

Proof: Suppose this is not true. Then there is a fraction $> \frac{1}{d}$ of constraint blocks whose constraints are in C', that are covered.

The expected number of constraints in C' that are satisfied by the assignment chosen in Lemma 1 is:

$$\sum_{z \in Z'} |P(z)| \cdot \frac{1}{2t^2(d-2)^t}$$

Since $\sum_{z \in Z'} |P(z)| \geq \frac{|C'|}{d}$, and C' contains a fraction $\geq (1 - 8\delta)$ of all the constraints, $\delta \leq \frac{1}{10}$, the total fraction of satisfied constraints is:

$$\frac{(1-8\delta)}{2dt^2(d-2)^t} \geq \frac{1}{10dt^2(d-2)^t} \geq \frac{1}{d^{2t}} = \frac{1}{d^{2d^5}} \geq \frac{1}{2^{d^6}}$$

Since $\ell = d^7$, and $d > \frac{1}{\alpha}$, this fraction is $\geq 2^{-\alpha\ell}$, which contradicts that φ is a no-instance. □

3.4 Conclusion

Our basic Set Cover construction has two parameters d, b. It has the following properties:

- In yes-instance: a fraction $\frac{1}{d-2} \leq \frac{2}{d}$ of sets is enough to cover all the elements.
- In no-instance: if we use a fraction $\leq \delta$ of basic set blocks, $\delta \leq \frac{1}{10}$, there is a fraction $\leq 8\delta + \frac{1}{d}$ of constraint blocks of elements which are covered.

Let M denote the number of basic set blocks in our construction, and let N denote the number of constraint blocks of elements in our construction. Since $M = 2n^l \cdot (d-2)^{8^\ell}$, and $N = (3n)^\ell$, $M \geq N$.

We define a new function $F(x) = 2^{2^x}$. Let B denote the size of a constraint block. Note that BN is the number of elements in our construction, further

$$B \leq 2^{d \cdot 8^\ell} \cdot b^d \cdot d^{d \cdot 8^\ell} \leq b^d \cdot d^{2d \cdot 8^\ell} \leq b^d \cdot 2^{2^{3\ell d}} = b^d F(3d^8) \leq b^d F(d^9)$$

4 The Reduction of GAP3SAT(3) to Asymmetric k-center

Given an instance φ of 3SAT(3), we build a directed graph with $h+1$ layers of vertices. For each pair of consecutive layers i, $(i+1)$, there are directed edges from some layer i vertices to some layer $(i+1)$ vertices. This graph is translated into an instance of k-center as follows. The set of vertices remains the same. For some i, consider layer i vertex v and layer $(i+1)$ vertex u. The distance $c(u,v) = \infty$. If there is an edge (v,u) in the original graph, then $c(v,u) = 1$. Otherwise, $c(v,u) = \infty$. The distances between any other pair of vertices are implied by these distances.

Layer 0 of vertices consists of only one vertex, which is connected to each vertex in layer 1. For each pair of consecutive layers $(i, i+1)$, $1 \leq i < h$, we build a Set Cover instance denoted by SC_i. In this Set Cover instance, vertices of layer i are viewed as sets and vertices of layer $(i+1)$ are viewed as elements. Thus, there are $(h-1)$ Set Cover instances in our construction. The edges are directed from layer i towards layer $i+1$ for all i.

The vertices of each layer are divided into blocks. The size of block in layer i is b_i, and $b_1 = 1$.

Layer i Set Cover instance, SC_i, consists of c_i copies of the basic Set Cover instance with parameters d_i, b_i. Layer i blocks of vertices are viewed as basic set blocks of SC_i. The constraint blocks of SC_i

become layer $i+1$ blocks of vertices, which is also the basic set blocks of SC_{i+1}. To achieve this, we need to fix the sizes b_i of blocks appropriately.

For a basic Set Cover instance with parameter d_i, denote by M_i the number of basic set blocks and N_i the number of constraint blocks. We must make sure that $c_i M_i = c_{i-1} N_{i-1}$.

Recall that $F(x) = 2^{2^x}$. The function $\log^{(i)} n$ is defined as follows: $\log^{(1)} n = \log n$ and $\log^{(i+1)} n = \log(\log^{(i)} n)$. Similarly, we define a function $2^{(i)}$ as follows: $2^{(1)} = 2$ and $2^{(i+1)} = 2^{2^{(i)}}$.

Let $r = \left\lfloor \frac{\log^* n}{2} \right\rfloor$. We define the parameters d_i as follows. $d_1 = 2^{(r)} \approx log^{(r)} n$, and $d_{i+1} = F(2^{d_i})$. The number of layers h is fixed in such a way that $d_{h-1} \leq \log^{(3)} n$. Thus, $h = \left\lfloor \frac{\log^* n}{6} \right\rfloor = \Theta(\log^* n)$.

We fix the number of vertices in layer 1 to be $\prod_{j=1}^{h-1} M_j$, where each vertex is a block. Thus the number of blocks in layer 1 is $\prod_{j=1}^{h-1} M_j$. For layer 1 SC instance, SC_1, we use $c_1 = \prod_{j=2}^{h-1} M_j$ basic SC instances with parameters b_1, d_1. Recall that each such SC instance has M_1 basic set blocks and N_1 constraint blocks. Thus, in layer 2, the number of blocks becomes $N_1 \prod_{j=2}^{h-1} M_j$. We fix $c_2 = N_1 \prod_{j=3}^{h-1} M_i$, and so on.

In general, layer i contains $B_i = \prod_{j=1}^{i-1} N_j \cdot \prod_{j=i}^{h-1} M_j$ blocks. We set $c_i = \prod_{j=1}^{i-1} N_j \cdot \prod_{j=i+1}^{h-1} M_j$. It is easy to see that $c_i N_i = c_{i+1} M_{i+1}$.

We now bound the size of our construction.

Block sizes: In layer 1, each vertex is a block, i.e., $b_1 = 1$. From our previous computation, given b_i (the size of basic set blocks of SC_i), the size of constraint block is $b_{i+1} \leq F(d_i^9) b_i^{d_i}$

Claim 3 *For all $i > 1$, $b_i \leq F(d_{i-1}^{10})$.*

Proof: For $i = 2$, $b_2 \leq F(d_1^9)$ and the claim is true. Fix some $i > 2$.

$$b_{i+1} \leq F(d_i^9) b_i^{d_i} \leq F(d_i^9) \left(F(d_{i-1}^{10})\right)^{d_i} \leq F(d_i^9) d_i^{d_i} \leq F(d_i^{10})$$

□

Size of the construction: The number of vertices in layer $h-1$ is at most:

$$\prod_{j=1}^{h-1} N_j \cdot b_h \leq (N_{h-1})^{h-1} b_h \leq (3n)^{d_{h-1}^7 (h-1)} \cdot F(d_{h-1}^{10}) \leq n^{poly(\log\log n)} \cdot 2^{2^{\log\log n}} = n^{poly(\log\log n)}$$

(since $d_{h-1} \leq \log^{(3)} n$.)

The total number of vertices is $|V| \leq h n^{poly \log\log(n)} \leq n^{poly \log\log(n)}$. Note that $\log^* n = \Theta(\log^* |V|)$, and so $h = \Theta(\log^* V)$.

Theorem 5 (Yes-Instance) *Suppose φ is a yes-instance. Let V_i denote the number of vertices in layer i. Then using at most $k = 4V_1/d_1$ centers we can cover all vertices within distance* 1.

Proof: For the solution of SC_i, we use $k_i \le \frac{2V_i}{d_i}$ of layer i vertices. We will show by induction that $k_i \le 2V_1/d_1^i$.

Clearly, the claim is true for $i = 1$. Assume it is true for $i-1$, i.e., $k_{i-1} \le 2V_1/d_1^{i-1}$. It is enough to show that $k_i \le \frac{k_{i-1}}{d_1}$. Then

$$\begin{aligned} k_i &= \frac{2V_i}{d_i} \le \frac{2V_{i-1}}{d_{i-1}} \cdot \frac{V_i}{V_{i-1}} \cdot \frac{d_{i-1}}{d_i} \\ &\le k_{i-1} b_i \cdot \frac{N_{i-1}}{M_{i-1}} \cdot \frac{d_{i-1}}{d_i} \\ &\le k_{i-1} F(d_{i-1}^{10}) \cdot d_{i-1} \cdot d_1/d_i d_1 \qquad (\text{since } d_i = F(2^{d_{i-1}})) \\ &< k_{i-1}/d_1 \end{aligned}$$

Therefore, the total number of vertices we use in the solution is $k = \sum_i k_i \le (1 + \frac{1}{d_1})k_1 \le 4V_1/d_1$.
□

Note that the fraction of vertices used in the layer corresponds to allocating $\frac{1}{d_i}$ centers fractionally on every vertex to cover all the elements in the next layer. This points to an Integrality Gap, and is demonstrated explicitly in the Appendix C.

Theorem 6 (No-Instance) *If the formula φ is a no-instance, then it is impossible to cover all the vertices with radius $h-1$, using $k+1$ centers.*

Proof: It is enough to prove, that it is impossible to cover all the vertices in layer h by k vertices in layer 1. Indeed, any solution must contain the vertex in layer 0 (as this is the only way to cover it). This vertex covers all the vertices within a radius of $h-1$, except for the layer h vertices. In order to cover layer h vertices, there is no point selecting centers in any layer other than 1: if we choose a center v in some layer $i > 1$, we can cover all the same vertices by choosing its predecessor in layer 1.

Define $\delta_1 = 4/d_1$, and $\delta_i = 20\delta_{i-1}$. Note that since d_i increases from layer to layer, $\delta_i \ge \frac{1}{d_i}$. Since $d_1 = 2^{(r)}$, then for all i, $\delta_i \le 20^{h-1} \cdot \frac{4}{2^{(r)}} \le \frac{1}{10}$ for sufficiently large n.

Lemma 3 *Suppose we have a solution that uses a fraction k of vertices in layer* 1*. Then for each layer i, the fraction of vertex blocks that are covered is $\le \delta_i$.*

Proof: By induction. For $i = 1$, the fraction of blocks in layer 1 that are in the solution is indeed δ_1.

Consider some general i, and assume the fraction of vertex blocks in layer i that are covered by the solution is $\le \delta_i$.

Consider the set cover instance SC_i. The fraction of basic set blocks that are covered is δ_i. The fraction of basic Set Cover instances in SC_i that contain more than a fraction $\frac{1}{10}$ of covered basic set blocks is at most $10\delta_i$.

Consider some other basic Set Cover instance in SC_i, and let $\delta \leq \frac{1}{10}$ be the fraction of basic set blocks that are covered in this instance. Then the fraction of constraint blocks that are covered in this instance is at most $8\delta + \frac{1}{d_i}$.

In total, the fraction of constraint blocks that are covered in layer $i+1$ is at most $8\delta_i + \frac{1}{d_i} + 10\delta_i \leq 20\delta_i = \delta_{i+1}$. □

□

Theorem 7 *There is no constant factor approximation for the asymmetric k-center problem unless $P = NP$ (use our construction with a constant h). There is no $c \log^* n$ (also, $c \log^* k$) approximation for some constant c, unless $NP \subseteq DTIME(n^{poly \log\log(n)})$.*

We remark that our results also holds for weaker assumptions. In fact, we prove the following. For every constant i, there is no $c_i \log^* n$-approximation (for some constant c_i), unless $NP \subseteq DTIME(n^{\log^{(i)} n})$. To see this, use our construction, choosing h to be the largest integer, such that $d_{h-1} \leq \log^{(i-2)} n$. Clearly, $h = \Theta(\log^* n)$, and the size of the construction is bounded by $n^{\log^{(i)} n}$.

5 Implications for Symmetric Distance Functions

The same reduction for $h = 3$ shows several other interesting hardness results for the weighted versions of metric k-center problem.

The weighted metric k-center is defined as follows: we are given a distance metric c over the vertices and a weight function w for the vertices. We want to choose a subset S of vertices of weight at most K so as to minimize

$$\max_{j \in V} \min_{i \in S} c_{ij}$$

where K is part of the input.

The problem is the familiar metric k-center problem for $K = k$ and $w(i) = 1$ for all i. The weighted metric k-center problem has a factor 3 approximation algorithm and in what follows we show that the result is tight.

Theorem 8 *It is NP-Hard to approximate the weighted metric k-center to a factor less than 3.*

Proof: We will construct the same layered instance as in the asymmetric case, but with $h = 3$ and d_1 a sufficiently large constant. Since the number of layers are constant the instance can be constructed in polytime. The edges in this case are however undirected.

The vertices in layer 3 have arbitrarily large weight (greater than K suffices) to rule out choosing them in any solution. The weight of all other vertices is 1.

In the case of the formula φ being a yes instance, we can cover all vertices within distance 1 using at most $4V_1/d_1$ vertices from layers $0, 1$ and 2.

If the formula were no instance, allocating the entire budget to the vertices in layer 1 we cannot cover all the vertices in layer 3. This means that in the case of a no-instance, we cannot cover all the vertices within distance less than 3. □

References

[1] A. F. Archer. Two $O(log^*k)$-approximation algorithms for the asymmetric k-center problem. *Proceedings of the 8th Conference on Integer Programming and Combinatorial Optimization*, pages 1–14, 2001.

[2] S. Arora and C. Lund. Hardness of approximation. In Dorit Hochbaum, editor, *Approximation Algorithms for NP Hard Problems.* PWS publishing Co, 1996.

[3] I. Dinur, V. Guruswami, and S. Khot. Vertex cover on k-uniform hypergraphs is hard to approximate within factor $(k-3-\epsilon)$. *Electronic Colloquium on Computational Complexity (ECCC)*, (027), 2002.

[4] I. Dinur, V. Guruswami, S. Khot, and O. Regev. A new multilayered PCP and the hardness of hypergraph vertex cover. *Proceedings of the* 35^{th} *ACM Symposium on Theory of Computing*, 2003.

[5] I. Dinur, O. Regev, and C. D. Smyth. The hardness of 3-uniform hypergraph coloring. *Proceedings of the* 43^{rd} *IEEE conference on Foundations of Computer Science*, 2002.

[6] I. Dinur and S. Safra. The importance of being biased. *Proceedings of the* 34^{th} *ACM Symposium on Theory of Computing*, 2002.

[7] E. Halperin, G. Kortsarz, and R. Krauthgamer. Personal communication. 2003.

[8] E. Halperin and R. Krauthgamer. Polylogarithmic inapproximability. *Proceedings of the* 35^{th} *ACM Symposium on Theory of Computing*, 2003.

[9] D. S. Hochbaum and D. B. Shmoys. A unified approach to approximation algorithms for bottleneck problems. *JACM*, 33(3):533–550, 1986.

[10] R. Panigrahy and S. Vishwanathan. An $O(log^*n)$ approximation algorithm for the asymmetric p-center problem. *J. of Algorithms*, 27(2):259–268, 1998.

[11] Vijay Vazirani. *Approximation Algorithms.* Springer Verlag, 2001.

Appendix

A Intersecting Families

This is exactly the same as appeared in [4], with one difference: we need to perform the calculation of their result in the central lemma.

Suppose we are given some basic set $U = [N] = \{0, 1, \ldots, N-1\}$. We look at the families of subsets of U.

Definition: A family of sets is called (d,t)-intersecting, if for every d sets in the family, the size of their intersection is at least t.

Each subset $S \subseteq U$ has a weight $w(S)$ (also called normalized weight). This weight is exactly the probability of choosing subset S, where each element of U is chosen independently with probability p. Thus, $w(S) = p^{|S|}(1-p)^{N-|S|}$. The weight of a family $\mathcal{F}$ is the sum of weights of all the sets in the family, and is exactly the probability of choosing one of the sets in $\mathcal{F}$.

Definition: Let $\mathcal{F}$ be a $(d-1,t)$-intersecting family. We say that $\mathcal{F}$ has the prefix property, if for each subset $S \in \mathcal{F}$, for some j, $|S \cap [t + (d-1)j]| \geq (d-2)j + t$. In this case we say that S has j-prefix property.

Lemma 4 *Let $\mathcal{F}$ be a $(d-1,t)$-intersecting family. Then there is another $(d-1,t)$-intersecting family $\mathcal{F}'$ of the same weight as $\mathcal{F}$, that has the prefix property.*

The following lemma is the central lemma.

Lemma 5 *Let $t = d^5$. Let $\mathcal{F}$ be a $(d-1,t)$-intersecting family with weight $\geq \frac{1}{4}$. Then there are $d-1$ sets in $\mathcal{F}$ whose intersection size is less than t.*

Proof: Suppose the claim is not true. $\mathcal{F}$ is a $(d-1,t)$-intersecting family of weight $\geq \frac{1}{4}$. Let $\mathcal{F}'$ be the corresponding family from previous lemma. The weight of $\mathcal{F}'$ is also $\geq \frac{1}{4}$, and it has the prefix property. The weight of $\mathcal{F}'$ is bounded by the probability of choosing some $S \in \mathcal{F}'$, which is bounded by the probability of choosing any subset with a prefix property. Therefore, the probability to choose a subset with a prefix property must be $\geq \frac{1}{4}$. We show that this is not true.

For some $j \geq 0$, we bound the probability of choosing a set with j-prefix property. Let x_i, $0 \leq i \leq t + (d-1)j - 1$ be the random variable indicating whether i is chosen. Clearly, $x_i = 1$ with probability p. Let $X = \sum x_i$.

Recall that $p = 1 - \frac{1}{d-2}$. Denote $\delta = \frac{d-2}{d-1} - p = \frac{1}{(d-1)(d-2)}$. Observe that $\mu = E[X] = p(t + (d-1)j)$.

We use Chernoff bound that says that if $\delta \leq \frac{1}{2}$, then:

$$Pr[X \geq \mu(1+\delta)] \leq e^{-\frac{\delta^2}{4}\mu}$$

Therefore,

$$Pr = Pr[X \geq t + (d-2)j] \leq Pr[(X-\mu) \geq \mu\delta/p] \leq e^{-\delta^2\mu/4p^2} = e^{-\delta^2(t+(d-1)j)/4p}$$

Summing over all j:

$$Pr \leq \sum_j e^{-\delta^2(t+(d-1)j)/4p} = \frac{e^{-\delta^2 t/4p}}{1 - e^{-(d-1)\delta^2/4p}}$$

As for all x: $0 < x \leq \frac{1}{2}$, $1 - e^{-x} \geq \frac{x}{2}$, and $(d-1)\delta^2/4p \leq \frac{1}{2}$,

$$1 - e^{-(d-1)\delta^2/4p} \geq (d-1)\delta^2/8p = \frac{d-1}{8}\frac{d-2}{d-3}\frac{1}{(d-1)^2(d-2)^2} \geq \frac{1}{8d^3}$$

Therefore,

$$Pr \leq 8d^3 e^{-\delta^2 t/4p}$$

Replacing $t = d^5$, we get:

$$\delta^2 t/4p = \frac{1}{4(d-1)^2(d-2)^2} \cdot \frac{d-2}{d-3} \cdot d^5 \geq \frac{d}{4}$$

and

$$Pr \leq 8d^3 e^{-\frac{d}{4}} < \frac{1}{4}$$

Contradicting the assumption that the weight of $\mathcal{F}'$ is $\geq \frac{1}{4}$.

□

B Proof of Lemma 1

Proof: (of Lemma 1) Given $z \in Z'$, let $y \in P(z)$.

Let $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1} \in I(y)$ be the subsets of assignments to y that determine $T(y)$, and let $\mathcal{B} \in I(z)$ be some subset of assignments to z. Note that none of the assignment subsets $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1}, \mathcal{B}$ are not chosen (by the definition of $I(y)$ and $I(z)$), and yet the constraint $C(y, z)$ is covered. This means that there is no basic element block corresponding to $\mathcal{A}_1, \ldots, \mathcal{A}_{d-1}, \mathcal{B}$, which means that there are some assignments $u \in \mathcal{B}$, $a \in T(y)$ that are consistent.

Consider now z and all the variables $y \in P(z)$. We denote by $T'(y)$ the set of assignments to z that are consistent with the assignments in $T(y)$. Note that since $|T(y)| < t$, $|T'(y)| < t$. For each $\mathcal{B} \in I(z)$ and for each $y \in P(z)$, there is some $u \in \mathcal{B}$, such that $u \in T'(y)$.

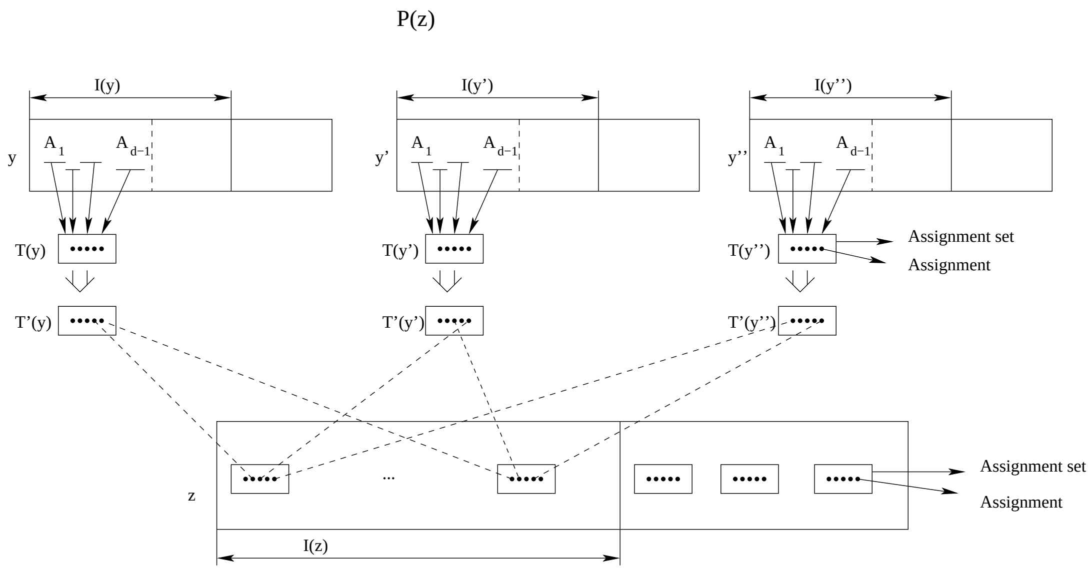

Figure 3: z and $P(z)$

Let q be the number of disjoint sets $T'(y)$, $y \in P(z)$. The rest of the proof consists of two claims. In the first claim, we show that q must be small (otherwise, it is impossible that the weight of z is $\geq \frac{1}{4}$.) In the second claim, we show that since there is a small number of disjoint sets $T'(y)$, there must be at least one assignment u that participates in many such sets.

Claim 4 $q \leq 2(d-2)^t$

Proof:

Each $\mathcal{B} \in I(z)$ must contain a distinct assignment belonging to each one of the q disjoint sets $T'(y)$. The normalized weight of $I(z)$ is the probability to choose one of the subsets of assignments in $I(z)$. This probability is less than or equal to the probability that we choose, for each $y \in P(z)$, at least one assignment in $T'(y)$. As $|T'(y)| < t$, the probability of choosing one of these assignments is $< 1-(1-p)^t$.

Therefore, the normalized weight of $I(z)$ is $< \left(1-(1-p)^t\right)^q$, and we know that it is at least $\frac{1}{4}$.

Assume that $q > 2(d-2)^t$. Then the weight of $I(z)$ is at most:

$$\left(1-(1-p)^t\right)^{2(d-2)^t} = \left(1 - \frac{1}{(d-2)^t}\right)^{2(d-2)^t} < e^{-2} < \frac{1}{4}$$

A contradiction. $\square$

Claim 5 *Suppose we are given a family of n sets of sizes $< m$, and each element appears in at most k sets. Then there are more than $\frac{n}{mk}$ disjoint sets in this family.*

Proof: By induction on the number of sets n. If $n = 1$, the proof is trivially true.

Consider some general n. Take out one of the sets of the family S. The number of sets intersecting with S (including S) is less than mk. Remove all these sets. By induction hypothesis, the remaining family has more than $\frac{n-mk}{mk}$ disjoint sets. Now add the set S to the collection of disjoint sets. $\Box$

Suppose there is no assignment u to z with $count(u) \geq \frac{1}{2t(d-2)^t} \cdot |P(z)| = k$. Consider the sets $T'(y)$ for $y \in P(z)$. We have $|P(z)|$ sets of less than t assignments, and each assignment appears in less than k sets. Then, by Claim 5, we must have more than $\frac{|P(z)|}{tk} = 2(d-2)^t = q$ disjoint sets which contradicts Claim 4 $\Box$

C An Explicit Integrality Gap

We consider the following LP:

$$
\begin{array}{lll}
\min & r & \\
\text{s.t.} & & \\
& \sum_v x_v = k & \\
& \sum_v y_{vu} = 1 & \forall u \in V \\
& y_{vu} \leq x_v & \forall v, u \in V \\
& y_{vu} c(v,u) \leq r & \forall v, u \in V
\end{array}
$$

We remark that our integrality gap construction works as well for the LP considered in [1].

Our construction has the same structure as before. We have $h+1$ layers of vertices, $h = \Theta(\log^* n)$, where n is the total number of vertices. Layer 0 contains only one vertex, which is connected to all the vertices in layer 1. For all other pairs of consecutive layers, i, $i+1$, we build a set cover instance SC_i, where layer i vertices serve as sets, and layer $i+1$ vertices are elements. The translation of this directed graph into an instance of k-center is performed as previously. However, our set cover instances SC_i are constructed differently.

Let m_i denote the number of vertices in layer i, and let $m = m_1 \approx \sqrt{n}$. Set cover instance SC_i has two parameters s_i and c_i and is constructed as follows. We divide all the sets in layer i into m_i/s_i disjoint collections of s_i sets. Let S be some such a collection of s_i sets. For each c_i-tuple of sets in S, there is an element in layer $i+1$, which is covered by these sets, and only by them.

The recursive formula for s_i is: $s_1 = 300$, and for $i > 1$, $s_{i+1} = 2^{(2^{s_i})}$.

The values of c_i are: $c_1 = 100$, and for $i > 1$, $c_{i+1} = 2^{s_i}$.

Observe that the number of sets increases from layer to layer. We start with $m \approx \sqrt{n}$ sets, and the recursive formula is: $m_{i+1} = \frac{m_i}{s_i} \cdot \binom{s_i}{c_i}$. Therefore, for each i, $m_i \geq m$. We choose h to be the largest possible integer, such that $m > 2^{s_h}$.

The Fractional Solution

In the fractional solution, the vertex at level 0 gets value 1. For layer i: each vertex gets a value of $\frac{1}{c_i}$. It is easy to see that this solution is feasible. Consider layer $i+1$ vertices By the construction, each element corresponding to any such vertex is covered by exactly c_i sets. Therefore, with weight $\frac{1}{c_i}$ on each layer i vertices, all the vertices at layer $i+1$ are completely covered.

Now we compute total fraction of open centers used by the fractional solution (k), which is the maximal number of centers we are allowed to use in any integral solution.

We open one center in layer 0. In layer 1, the total sum of fractions on the vertices is $\frac{m}{100}$. We show that for each i, $i > 1$, the value of the fractional solution at layer i is less than $\frac{m}{100 \cdot 2^{i-1}}$. Therefore, $k \leq \frac{m}{50}$.

By the recursion formula,

$$\begin{aligned}
m_i &= m \prod_{j<i} \frac{\binom{s_j}{c_j}}{s_j} \\
&\leq \frac{m}{s_1} \prod \frac{s_j^{c_j}}{2} \\
&\leq \frac{m}{100 \cdot 2^{i-1}} \cdot \left(s_i^{c_i}\right)^i \\
&\leq \frac{m}{100 \cdot 2^{i-1}} \cdot \left(s_i\right)^{c_i^2} \\
&= \frac{m}{100 \cdot 2^{i-1}} \cdot 2^{2^{s_i - 1} \cdot 2^{2 s_i - 1}} \\
&\leq \frac{m}{100 \cdot 2^{i-1}} \cdot 2^{(2^{3 s_i - 1})}
\end{aligned}$$

Observe that $s_i = 2^{(2^{s_i-1})} > (2^{3s_{i-1}})$. Therefore, $m_i \leq \frac{m}{100 \cdot 2^{i-1}} \cdot 2^{s_i} = \frac{m}{100 \cdot 2^{i-1}} \cdot c_i$. Since the value of the fractional solution at layer i is exactly $\frac{m_i}{c_i}$, we have that this value is at most $\frac{m}{100 \cdot 2^{i-1}}$. In total, the fractional solution uses at most $\frac{m}{50}$ centers.

The Integral Solution

We show that it is impossible to cover all the vertices with $k = \frac{m}{50}$ centers within the radius $h-1$. Again, it is enough to show that choosing any subset of k layer 1 vertices leaves some layer h vertices uncovered.

Fix some such solution. For each i, $i \geq 1$, let α_i be such that the fraction of vertices that are not covered at layer i is $\frac{1}{\alpha_i}$. For example, $\alpha_1 = \frac{50}{49}$. We call such vertices, that do not have ancestors at layer 1 which are in the solution, "good vertices". We prove the following lemma:

Lemma 6 *For all $i > 1$, $\alpha_i \leq 2^{s_{i-1}}$.*

Observe that if this lemma is true, then the number of vertices in layer r that are not covered is at least $m/2^{s_{h-1}} > 1$. Therefore, the proof of this lemma will complete the proof of the integrality gap.

Proof: The proof is by induction. We can define $s_0 = \log\log s_1 > 1$, which is consistent with the recursive definition of s_i. Since $\alpha_1 = 50/19 < 1$, $\alpha_1 < 2^{s_0}$, and the claim is true for $i = 1$.

Assume the claim is true for i. We prove it for $i+1$. First, observe that $s_i = 2^{2^{s_{i-1}}}$, and by induction hypothesis, $\alpha_i \leq 2^{s_{i-1}}$. Also, recall that $c_i = 2^{s_{i-1}}$. Therefore, $s_i >> \alpha_i$, and also $s_i >> \alpha_i c_i$. The number of good vertices in layer i is at least $\frac{m_i}{\alpha_i}$. In the set cover instance for layer i, all the level i vertices are divided into m_i/s_i collections of s_i vertices each.

What is the smallest possible number of collections that have less than $\frac{s_i}{2\alpha_i}$ good vertices? In the worst case, we put $\frac{s_i}{2\alpha_i} - 1$ good vertices in each set, and pack the remaining $\frac{m_i}{2\alpha_i}$ vertices into smallest possible number of sets, which is $\frac{m_i}{2\alpha_i s_i}$. Therefore, at least a fraction $\frac{1}{2\alpha_i}$ of collections has more than $\frac{s_i}{2\alpha_i}$ good vertices (It is important here that $s > 2\alpha_i$, so that this number is at least 1). Consider any such collection S of vertices. We denote $b_i = 2\alpha_i$. Therefore, the fraction of good elements in S is at least $\frac{1}{b_i}$. Let S' be the corresponding collection of vertices in layer $i+1$ (i.e., all the vertices that have directed edges from vertices in S). We denote by b_{i+1} the number such that the fraction of good vertices in S' is at least $\frac{1}{b_{i+1}}$. Then the total fraction of good vertices in layer $i+1$ is at least $\frac{1}{2\alpha_i b_{i+1}}$. Therefore, $\alpha_{i+1} \leq 2\alpha_i b_{i+1}$.

The next claim states that $b_{i+1} \leq e^{b_i c_i}$.

To complete the proof of the lemma,

$$\begin{aligned}
\alpha_{i+1} &\leq 2\alpha_i b_{i+1} \\
&\leq 2\alpha_i e^{b_i c_i} \\
&= 2\alpha_i e^{2\alpha_i c_i} \\
&\leq 2^{8\alpha_i c_i} \\
&\leq 2^{8\cdot 2^{2s_{i-1}}} \\
&\leq 2^{(2^{2s_{i-1}}+3)}
\end{aligned}$$

We want to prove that $\alpha_{i+1} \leq 2^{s_i}$. It is enough to see that $(2^{2s_{i-1}} + 3) \leq s_i = 2^{(2^{s_{i-1}})}$, which is clearly true.

Finally, we prove the following claim:

Claim 6 $b_{i+1} \leq e^{b_i c_i}$.

Proof: For the sake of convenience, in this proof only, we denote $s_i = s$, $b_i = b$, $c_i = c$. The fraction of good vertices in S' is:

$$
\begin{aligned}
\frac{1}{b_{i+1}} &= \frac{\binom{s/b}{c}}{\binom{s}{c}} \\
&= \frac{(s-c)!}{(s/b-c)!} \cdot \frac{(s/b)!}{s!} \\
&\geq \left(\frac{s/b-c}{s/b}\right)^{s(1-1/b)} \\
&\geq \left(1 - \frac{1}{s/bc}\right)^{s} \\
&\approx e^{-bc}
\end{aligned}
$$

The latter is true since $s >> bc$. □

□

ECCC ISSN 1433-8092
http://www.eccc.uni-trier.de/eccc
ftp://ftp.eccc.uni-trier.de/pub/eccc
ftpmail@ftp.eccc.uni-trier.de, subject 'help eccc'