SIAM J. COMPUT.
Vol. 36, No. 2, pp. 498–515


# THE HARDNESS OF METRIC LABELING*

JULIA CHUZHOY† AND JOSEPH (SEFFI) NAOR‡

**Abstract.** The metric labeling problem is an elegant and powerful mathematical model capturing a wide range of classification problems. The input to the problem consists of a set $L$ of labels and a weighted graph $G = (V, E)$. Additionally, a metric distance function on the labels is defined, and for each label and each vertex, an assignment cost is given. The goal is to find a minimum-cost assignment of the vertices to the labels. The cost of the solution consists of two parts: the assignment costs of the vertices and the separation costs of the edges (where each edge pays its weight times the distance between the two labels to which its endpoints are assigned).

Due to the simple structure and the variety of applications, the problem and its special cases (with various distance functions on the labels) have recently received much attention. Metric labeling is known to have a logarithmic approximation, and it has been an open question for some time whether a constant approximation exists. We refute this possibility and prove that no constant factor approximation algorithm exists for metric labeling, unless P=NP. Moreover, we prove that the problem is $\Omega\left((\log |V|)^{1/2-\delta}\right)$-hard to approximate for any constant $\delta : 0 < \delta < 1/2$, unless NP has quasi-polynomial time algorithms.

**Key words.** metric labeling, 0-extension, markov random field, hardness of approximation.

**AMS subject classifications.** 68Q25, 68W25, 90C27, 90C59

**DOI.** 10.1137/S0097539703422479

## 1. Introduction.

The metric labeling problem, first formulated by Kleinberg and Tardos [18], captures a broad range of classification problems that arise in computer vision and related fields. In such classification problems, the goal is to assign labels to some given set of objects in a way consistent with observed data or some other form of prior knowledge. Formally, the input to the metric labeling problem consists of an $n$-vertex undirected graph $G(V, E)$ with weights $w$ on edges, and a set $L$ of labels with metric distance function $d : L \times L \to R$ associated with them. Additionally, for each vertex $v \in V$ and label $\ell \in L$, an assignment cost $c(v, \ell)$ is specified. The problem output is an assignment $f : V \to L$ of the vertices to the labels. Intuitively, the vertices are the objects we would like to classify, and the assignment function $f$ provides such a classification. The prior knowledge is modeled by the means of the vertex assignment costs $c(v, \ell)$, that can be used to express an estimate on how likely it is that $\ell$ is the correct label for vertex $v$, and by the edge weights, which define pairwise relations between the objects. The weights of the edges express a prior estimate on how likely it is that the endpoints of the given edge should be assigned to close or identical labels.

Given a solution $f : V \to L$ to the metric labeling problem, its cost $Q(f)$ consists of two components.

*Received by the editors February 9, 2003; accepted for publication (in revised form) February 2, 2006; published electronically July 31, 2006. A preliminary version of this work appeared in the *Proceedings of the 45rd Annual IEEE Symposium on Foundations of Computer Science*, Rome, Italy, 2004, pp. 108-114.

http://www.siam.org/journals/sicomp/36-2/42247.html

†School of Mathematics, IAS, 1 Einstein Drive, Princeton NJ 08540 (cjulia@csail.mit.edu). This work was done while this author was a graduate student at the Computer Science Department at the Technion.

‡Computer Science Department, Technion, Haifa 32000, Israel (naor@cs.technion.ac.il). This author's research was supported in part by US-Israel BSF grant 2002276 and by EU contract IST-1999-14084 (APPOL II).

**Vertex Labeling Cost:** For each $v \in V$, this cost is $c(v, f(v))$.
**Edge Separation Cost:** For each edge $e = (u, v)$ the cost is $w(u, v) \cdot d(f(u), f(v))$.

Thus,

$$Q(f) = \sum_{u \in V} c(u, f(u)) + \sum_{(u,v) \in E} w(u, v) \cdot d(f(u), f(v))$$

and the goal is to find a labeling $f : V \to L$ minimizing $Q(f)$.

Metric labeling has rich connections to some well known problems in combinatorial optimization. A special case of metric labeling is the *0-extension* problem, studied by Karzanov [16, 17]. In this problem, there are no assignment costs. However, the graph contains a set $\{t_1, \ldots, t_k\}$ of terminals, where the label of terminal $t_i$ is fixed in advance to $i$, while the non-terminals are free to be assigned to any of the labels. As in the metric labeling problem, a metric is defined on the set of labels. The cost of an assignment consists only of the *edge separation cost*. The 0-extension problem generalizes the well-studied *multiway cut* problem [9, 6, 14], which is defined exactly like 0-extension, except that the metric on the label set is the uniform metric.

The first approximation algorithm for the metric labeling problem was shown by Kleinberg and Tardos [18]. This algorithm uses the probabilistic tree embedding technique [4, 5], and its approximation factor is $O(\log k \log \log k)$, where $k$ denotes the number of labels in $L$. This bound was recently improved to $O(\log k)$ [11] and it is currently the best known approximation factor for the metric labeling problem. Some special cases of metric labeling, where the metric on the terminals belongs to some restricted class of metrics, were shown to have better approximation factors [18, 13, 8].

Chekuri et al.[8] proposed a natural linear programming formulation for the general metric labeling problem. A solution to this linear program is an embedding of the graph in a $k$-dimensional simplex, where the distance between points in the simplex is defined by a special metric, the *earth mover's distance metric* (EMD), and not by the (standard) $\ell_1$ metric. It was also shown in [8], that the integrality gap of this formulation is at most the distortion of a probabilistic tree embedding of the given metric $d$, i.e., $O(\log k)$ [11]. Archer et al. [1] presented an approximation algorithm which is based on rounding the EMD solution to the linear program of [8] and achieves an $O(\log |V|)$ approximation factor.

Călinescu et al.[7] considered approximation algorithms for the 0-extension problem via a metric relaxation, originally studied by Karzanov [16], and obtained an $O(\log k)$-approximation algorithm for general metrics. This result was improved to $O(\log k / \log \log k)$ by Fakcharoenphol et al. [10].

*Our Results.* A question that has intrigued many researchers since the appearance of [18] is whether there exists a constant factor approximation algorithm for the metric labeling problem. We answer this question in the negative, and prove an $\Omega\left((\log n)^{1/2-\delta}\right)$-hardness of approximation for any constant $\delta : 0 < \delta < 1/2$, assuming NP $\not\subseteq$ DTIME $\left(n^{\mathrm{poly}(\log n)}\right)$. We also prove that there is no constant factor approximation algorithm for metric labeling unless P = NP. For the sake of simplicity, we focus on a problem called *restricted metric labeling*, which was shown by Chekuri et al.[8] to be equivalent to metric labeling. In the restricted metric labeling problem, the assignment costs of the vertices are restricted to be either 0 or $\infty$, or equivalently, each vertex $v \in V$ has a list of labels, $L(v)$, to which it is allowed to be assigned. The solution cost then only consists of the edge separation cost.

Following our work, Karloff et al. [15] showed that the 0-extension problem

is $\Omega\left((\log n)^{1/4-\epsilon}\right)$-hard to approximate for any constant $\epsilon$, unless $NP$ has quasi-polynomial time algorithms. Their proof builds on ideas presented in this work.

*Organization.* We start in Section 2 with some preliminaries, and we present in Section 3 a simple $(3-\delta)$-hardness proof (for any constant $0 < \delta < 1$) for the restricted metric labeling problem. This proof provides the intuition and motivation for the new ideas needed to obtain the stronger hardness bounds shown in Section 4.

**2. Preliminaries.** We prove our hardness results for the restricted metric labeling problem, defined as follows. The input consists of an undirected graph $G(V, E)$ with weights $w$ on edges, and a set $L$ of labels with distance function $d : L \times L \to R$. Additionally, for each vertex $v \in V$, there is a subset $L(v) \subseteq L$ of labels to which $v$ can be assigned. The goal is to find an assignment $f : V \to L$, such that for each $v \in V$, $f(v) \in L(v)$. The solution cost is the sum, over all edges $e = (u, v)$, of the edge separation cost $w(e)d(f(u), f(v))$. We notice that our hardness results work even for the uniform weight function, i.e., for each edge $e \in E$, $w(e) = 1$.

We perform our reduction from the gap version of Max 3SAT(5). The input to the problem is a CNF formula $\phi$ with $n$ variables and $\frac{5n}{3}$ clauses. Each clause consists of 3 literals and each variable participates in 5 clauses, appearing in each clause at most once.

Let $\epsilon : 0 < \epsilon < 1$, be a constant and let $\phi$ be an instance of Max 3SAT(5). Then $\phi$ is called a *Yes-instance* if there is an assignment that satisfies all the clauses, and it is called a *No-instance* (with respect to $\epsilon$) if any assignment satisfies at most a fraction $(1-\epsilon)$ of the clauses. Following is one of the several equivalent statements of the PCP theorem [2, 3].

THEOREM 2.1. *There is a constant $\epsilon$, $0 < \epsilon < 1$, such that it is NP-hard to distinguish between Yes-instances and No-instances of the Max 3SAT(5) problem.*

In our reduction, we start from a 3SAT(5) formula $\phi$, and produce an instance of the restricted metric labeling problem. Our first step is describing and analyzing a (standard) two-prover protocol for the 3SAT(5) problem. This protocol will help us translate 3SAT(5) instances into instances of restricted metric labeling, and analyze the reduction.

The one-round two-prover protocol for 3SAT(5) is defined as follows. Given a 3SAT(5) formula $\phi$ on $n$ variables:

- The verifier randomly chooses a clause $C$ from the formula $\phi$ and one of the variables $x$ belonging to $C$. Variable $x$ is called the *distinguished variable*.
- Prover 1 receives clause $C$ and is expected to return an assignment to all the variables appearing in the clause. Prover 2 receives variable $x$ and is expected to return an assignment to $x$.
- After receiving the answers of the provers, the verifier checks that the answer of prover 1 defines a satisfying assignment to clause $C$ and that the assignments of prover 1 and prover 2 to variable $x$ are identical.

The following well known theorem easily follows from Theorem 2.1.

THEOREM 2.2. *If $\phi$ is a Yes-instance, then there is a strategy of the provers such that the verifier always accepts. If $\phi$ is a No-instance, then for any strategy of the provers, the acceptance probability is at most $(1 - \frac{\epsilon}{3})$.*

**3. A Simple $(3 - \delta)$ Hardness.** In this section we present a simple $(3 - \delta)$-hardness for the restricted metric labeling problem (for any constant $0 < \delta < 1$), and also provide some intuition as to the new ideas needed to improve this bound.

We start by amplifying the soundness of the 2-prover protocol presented above by means of parallel repetitions, a usual practice in PCP reductions. The number of

repetitions is a sufficiently large constant $l$. The new protocol proceeds as follows.

- The verifier chooses, randomly and independently, $l$ clauses $C_1, \ldots, C_l$ from the input formula $\phi$. For each $i$, $1 \leq i \leq l$, the verifier chooses, randomly and independently, one variable $x_i$ belonging to $C_i$.
- Prover 1 receives clauses $C_1, \ldots, C_l$ and is expected to return an assignment to all the variables appearing in the clauses, such that all clauses are satisfied. Prover 2 receives variables $x_1, \ldots, x_l$ and is expected to return an assignment to these variables.
- After receiving the answers of the provers, the verifier checks that the answer of prover 1 defines satisfying assignments to clauses $C_1, \ldots, C_l$ and that the assignments of prover 1 and prover 2 to variables $x_1, \ldots, x_l$ are identical.

The following theorem follows from the well known Raz parallel repetition theorem [19], which bounds the error probability of the above protocol.

THEOREM 3.1. *There is a constant $\alpha > 0$, such that if $\phi$ is a Yes-instance, there is a strategy of the provers for which the verifier always accepts, and if $\phi$ is a No-instance, then for any strategy of the provers, the acceptance probability is at most $2^{-\alpha l}$.*

Let $Q_1$ denote the set of all the possible queries to prover 1 (i.e., each query $q \in Q_1$ is an $l$-tuple of clauses). Given a query $q_1 \in Q_1$, let $A(q_1)$ denote the set of all the assignments to the variables that appear in the clauses of $q_1$ that satisfy these clauses. Similarly, $Q_2$ denotes the set of all the possible queries to prover 2 (each query is an $l$-tuple of variables), and given $q_2 \in Q_2$, $A(q_2)$ is the set of all the possible answers of prover 2 to query $q_2$.

We assume that at the beginning of the protocol, the verifier chooses a random string $r$, which determines the choice of the clauses and the variables sent to the provers. Let $R$ denote the set of all the possible random strings. Given a random string $r \in R$, let $q_1(r), q_2(r)$ be the queries sent to prover 1 and prover 2 respectively, when the verifier chooses $r$.

The set of labels is defined as follows. For every possible query of each one of the two provers, and for every possible answer to this query, there is a label, i.e.,

$$L = \{\ell(q, A) \mid q \in Q_1 \cup Q_2, A \in A(q)\}$$

In order to define the metric distance function on the labels, we construct a label graph $G_L$. The vertices of this graph are the labels, and the metric distance between the labels is defined to be the length of the shortest path in this graph. We now define the edges of graph $G_L$. Consider some random string $r$ of the verifier, and let $q_1 = q_1(r)$, $q_2 = q_2(r)$. Let $A_1 \in A(q_1), A_2 \in A(q_2)$ be any pair of consistent answers to these queries. Then there is an edge of length 1 between $\ell(q_1, A_1)$ and $\ell(q_2, A_2)$ in $G_L$. Note that since each edge connects a label belonging to prover 1 and a label belonging to prover 2, the graph is bipartite. Therefore, for any random string $r$, if $q_1 = q_1(r)$ and $q_2 = q_2(r)$, and if $A_1 \in A(q_1)$, $A_2 \in A(q_2)$ are inconsistent answers to these queries, then the distance between labels $\ell(q_1, A_1)$ and $\ell(q_2, A_2)$ in graph $G_L$ is at least 3.

We now proceed to define the input graph. For every query $q \in Q_1 \cup Q_2$, there is a vertex $v(q)$. This vertex can only be assigned to those labels, that correspond to query $q$, i.e.,

$$V = \{v(q) \mid q \in Q_1 \cup Q_2\}$$

$$L(v(q)) = \{\ell(q, A) \mid A \in A(q)\}$$

The edge set is defined as follows. For each random string $r$ of the verifier, there is an edge connecting $v(q_1(r))$ and $v(q_2(r))$. All edges have unit weight.

*Yes-instance.* If $\phi$ is a Yes-instance, then there is a strategy of the provers such that their answers are always accepted by the verifier. This strategy defines the assignments of the vertices to the labels, namely, vertex $v(q)$ for $q \in Q_1 \cup Q_2$ is assigned to label $\ell(q, A)$, where $A \in A(q)$ is the answer of the corresponding prover to query $q$ under the above strategy. Consider some random string $r$ of the verifier and the corresponding queries $q_1 = q_1(r), q_2 = q_2(r)$. Let $A_1 \in A(q_1)$, $A_2 \in A(q_2)$ be the answers of the provers according to the above strategy. Note that vertices $v(q_1)$, $v(q_2)$ are assigned to labels $\ell(q_1, A_1)$, $\ell(q_2, A_2)$ and that the answers $A_1$ and $A_2$ of the provers are consistent. Therefore, there is an edge in the label graph between the labels $\ell(q_1, A_1)$ and $\ell(q_2, A_2)$, and thus the distance between the two labels (and the cost incurred by the edge connecting $v(q_1)$ and $v(q_2)$) is 1. The total cost of the solution is therefore $|R|$, where $R$ is the set of all the random strings of the verifier.

*No-instance.* Consider any solution to the problem. Note that the assignments of the vertices to the labels define a strategy of the provers (the assignment of vertex $v(q)$, $q \in Q_1 \cup Q_2$ to label $\ell(q, A)$, $A \in A(q)$, implies that the answer of the corresponding prover to query $q$ is $A$). Let $R' \subseteq R$ be the set of random strings of the verifier for which the answers of the two provers are inconsistent. From Theorem 3.1, $|R'| \geq (1 - 2^{-\alpha l})|R|$. Consider a random string $r \in R'$ and let $q_1 = q_1(r), q_2 = q_2(r)$. Let $\ell(q_1, A_1)$, $\ell(q_2, A_2)$ be the labels to which the vertices $v(q_1)$, $v(q_2)$ are assigned. As the answers $A_1$, $A_2$ of the provers are inconsistent, the distance between the two labels (and hence the separation cost paid by the edge between $v(q_1)$ and $v(q_2)$) is at least 3. Therefore, the total cost of the solution is at least $3(1 - 2^{-\alpha l})|R|) = 3(1 - \delta)|R|$, where $\delta$ is an arbitrarily small constant.

It follows that the gap between the costs of Yes and No instances is $3(1 - \delta)$, and since the size of the construction is polynomial in $n$, we have that restricted metric labeling is $3(1 - \delta)$-hard to approximate for any constant $\delta$, unless P=NP.

It is not hard to see that the analysis is tight. Consider some random string $r$ and the corresponding queries $q_1 = q_1(r)$, $q_2 = q_2(r)$. Let $A_1, A_2$ be a pair of inconsistent answers to queries $q_1, q_2$. We show that there is a path of length 3 in the graph $G_L$ between the pair of labels $\ell(q_1, A_1)$, $\ell(q_2, A_2)$. We denote $q_1 = (C_{i_1}, \ldots, C_{i_l})$ and $q_2 = (x_{i_1}, \ldots, x_{i_l})$, and recall that for each $j : 1 \leq j \leq l$, $x_{i_j}$ is one of the variables of clause $C_{i_j}$. Let $x'_{i_j}$ and $x''_{i_j}$ denote the other two variables. The path of length 3 connecting the two labels starts at label $\ell(q_1, A_1)$. The second label on this path is $\ell(q'_2, A'_2)$, where $q'_2 = (x'_{i_1}, \ldots, x'_{i_l})$ and $A'_2$ contains assignments to $(x'_{i_1}, \ldots, x'_{i_l})$ identical to those in $A_1$. The third label is $\ell(q_1, A'_1)$ (we define $A'_1$ below), and the final fourth label is $\ell(q_2, A_2)$. In order to define $A'_1$, fix some $j : 1 \leq j \leq l$, and consider the $j$th entry of $q_1$, a clause whose variables are $x_{i_j}, x'_{i_j}$ and $x''_{i_j}$. We need to specify the assignments to these variables in $A'_1$. The assignment to $x_{i_j}$ is defined to be the same assignment that appears in $A_2$, the assignment to $x'_{i_j}$ is the same as in $A'_2$, and the assignment to $x''_{i_j}$ is set in such a way that clause $C_{i_j}$ is satisfied.

Thus, even though the two answers $A_1$ and $A_2$ of the provers might be inconsistent in many coordinates, there is still a short path between the two labels. In order to improve the hardness bound, it would be helpful (and enough) to ensure that if two answers are inconsistent in almost all the coordinates, then the length of the shortest

path between the two corresponding labels is $\Omega(l)$. This is the intuition behind the construction and the $k$-prover protocol described in the next section.

**4. The Main Hardness Result.** In this section we prove an $\Omega\left((\log n)^{1/2-\delta}\right)$-hardness of restricted metric labeling, for any constant $\delta : 0 < \delta < \frac{1}{2}$. We start by defining a new $k$-prover protocol for 3SAT(5). The protocol is then used in a way which is similar to the construction in Section 3 to obtain the stronger hardness result.

**4.1. A New $k$-Prover Protocol.** We define a new $k$-prover protocol, where the $k$ provers are denoted by $P_1, \ldots, P_k$, and $k$ will be later set to poly$(\log n)$. This protocol is based on the basic one-round two-prover protocol, and it proceeds as follows.

- The verifier sends one query to each prover. Each one of the queries has $\binom{k}{2}$ entries, which are determined in the following way. For each pair $(i, j)$ of provers, where $1 \leq i < j \leq k$, the verifier chooses, uniformly, independently at random, a clause $C_{ij}$ and a distinguished variable $x_{ij}$ belonging to this clause. The entry $(i, j)$ in the queries of the provers is then defined as follows. In the query sent to prover $P_i$, this entry contains clause $C_{ij}$. In the query sent to prover $P_j$, this entry contains variable $x_{ij}$. For each other prover $P_a$, where $a \neq i, j$, the entry $(i, j)$ of its query contains both clause $C_{ij}$ and variable $x_{ij}$.
  Thus, in general, for any prover $P_h$, $1 \leq h \leq k$, coordinate $(y, z)$ of its query (where $1 \leq y < z \leq k$), is defined as follows:
  - if $h = y$, then the entry contains $C_{yz}$.
  - if $h = z$, then the entry contains $x_{yz}$.
  - if $h \neq y, z$, then the entry contains both $C_{yz}$ and $x_{yz}$.
- Each one of the provers responds with an assignment to all the variables appearing in its query, both as parts of clauses and as distinguished variables.
- After receiving the answers of the provers, the verifier checks, for each coordinate $(i, j)$, $1 \leq i < j \leq k$, that the answers of all the provers are consistent, i.e., all the provers $P_a$, $a \neq j$, return an identical assignment to the variables of $C_{ij}$, and the assignment of prover $P_j$ to variable $x_{ij}$ matches the assignments of all the other provers.

We note that our $k$-prover system departs from standard protocols in several ways. First, we do not use the parallel repetitions theorem here, as there is no need to amplify the soundness of the protocol. Observe also that for each prover $P_a$, for each coordinate $(i, j) : i, j \neq a$, the prover receives both the clause $C_{ij}$ and the distinguished variable $x_{ij}$. It may look that some of the information the prover receives is redundant. Indeed, in $k$-prover systems (e.g., [12]), the provers usually receive either the clause or the distinguished variable, but not both. However, this sending of redundant information to the provers is essential for our reduction. Intuitively, it will ensure that if, for some random string $r$, the answers of the $k$ provers are inconsistent in many coordinates, then the distances between the corresponding labels are long.

We assume again that all the random choices of the verifier are made at the beginning of the protocol, by choosing a random string $r$ out of the set $R$ of all the possible random strings of the desired length. Given a random string $r \in R$, for each $i$, $1 \leq i \leq k$, let $q_i(r)$ be the query sent to prover $P_i$ when the verifier chooses the random string $r$, and let $Q_i$ be the set of all the possible queries of prover $i$. For each $i : 1 \leq i \leq k$, for each $q_i \in Q_i$, let $A(q_i)$ denote the set of all the possible answers of prover $P_i$ to query $q_i$, which satisfy all the clauses appearing in the query.

DEFINITION 4.1. *Consider a pair of provers $P_i$ and $P_j$, $1 \le i < j \le k$, and let $q_i \in Q_i$, $q_j \in Q_j$ be a pair of queries, such that for some random string $r \in R$, $q_i = q_i(r), q_j = q_j(r)$. Let $A_i$ and $A_j$ denote the respective answers of the provers to the queries. We say that the answers are* weakly consistent *if the assignments to $C_{ij}$ and $x_{ij}$ in $A_i$ and $A_j$ respectively are consistent. The answers are called* strongly consistent *if they are also consistent in every other coordinate, i.e., for each $(a, b)$, $1 \le a < b \le k$, where $(a, b) \ne (i, j)$:*

- *If both entries $q_i(a, b)$ and $q_j(a, b)$ contain clause $C_{ab}$ and variable $x_{ab}$, then the assignments to the variables of clause $C_{ab}$ in $A_i$ and $A_j$ are identical.*
- *If one of the entries $q_i(a, b)$ and $q_j(a, b)$ contains clause $C_{ab}$ and the other contains clause $C_{ab}$ and variable $x_{ab}$, then the assignments to the variables of the clause $C_{ab}$ in $A_i$ and $A_j$ are identical.*
- *If one of the entries $q_i(a, b)$ and $q_j(a, b)$ contains variable $x_{ab}$ and the other contains clause $C_{ab}$ and variable $x_{ab}$, then the assignments to the variables of clause $C_{ab}$ and variable $x_{a,b}$ in $A_i$ and $A_j$ are consistent.*

THEOREM 4.2. *If $\phi$ is a Yes-instance, then there is a strategy of the $k$ provers such that the verifier always accepts. If $\phi$ is a No-instance, then for any strategy of the provers, for every pair of provers $P_i$ and $P_j$, $1 \le i < j \le k$, the probability that their answers are weakly consistent is at most $(1 - \frac{\epsilon}{3})$.*

*Proof.* For the Yes-instance, the theorem follows immediately. We now prove that the theorem holds for the No-Instance. Assume otherwise. Let $P_i$ and $P_j$ be a pair of provers such that the probability that their answers are weakly consistent is more than $(1 - \frac{\epsilon}{3})$. We partition the set of random strings $R$ into classes, such that within each class the random strings are identical except for the clause $C_{ij}$ and the distinguished variable $x_{ij}$. Each such class, (together with the corresponding queries and answers to the queries), can be viewed as a two-prover protocol (while we ignore all the coordinates of the queries and the answers except for $(i, j)$). As the probability of obtaining a pair of weakly consistent answers is more than $(1 - \frac{\epsilon}{3})$, at least for one of the classes, the probability that the verifier accepts is greater than $(1 - \frac{\epsilon}{3})$. This defines a strategy for the two-prover protocol in which the acceptance probability of the verifier is greater than $(1 - \frac{\epsilon}{3})$, contradicting Theorem 2.2. □

**4.2. The Graph and the Label Set.** In this section we construct an instance of the restricted metric labeling problem from an input 3SAT(5) formula $\phi$. Our construction is based on the $k$-prover system described above.

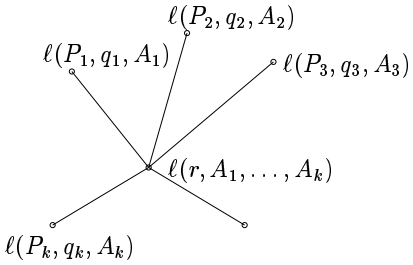


FIG. 4.1. *Edges in the graph of labels incident to $\ell(r, A_1, \ldots, A_k)$*

The set of labels $L$ consists of two subsets:

**Query Labels:** for each prover $P_i$, $1 \le i \le k$, for each query $q \in Q_i$, and for each answer $A \in A(q)$ to the query $q$, there is a label $\ell(P_i, q, A)$.

**Constraint Labels:** consider a random string $r$ of the verifier. Let $A_1, \ldots, A_k$, be any collection of possible answers of the provers to the queries $q_1(r), \ldots, q_k(r)$, i.e., for each $1 \leq i \leq k$, $A_i \in A(q_i(r))$. Moreover, assume that these answers are accepted by the verifier, (i.e., $A_1, \ldots, A_k$ are strongly consistent). Then, there is a label $\ell(r, A_1, A_2, \ldots, A_k)$.

We now define a graph $G_L(L, E')$ on the label set. The metric on the label set is implied by the shortest path distance function in the graph. The vertices of $G_L$ are the labels and the edges are defined as follows. Consider a constraint label $\ell = \ell(r, A_1, A_2, \ldots, A_k)$, Then, for each $i$, $1 \leq i \leq k$, there is an edge of length $\frac{1}{2}$ between $\ell$ and $\ell(P_i, q_i(r), A_i)$ (see Figure 4.1).

Thus, the graph is a collection of stars, while some stars share some of their leaves.

We now proceed to define graph $G(V, E)$. The vertex set $V$ is the union of two vertex sets: a set of *query vertices*, denoted by $V_1$, and a set of *constraint vertices*, denoted by $V_2$.

**Query Vertices:** for each prover $P_i$, $1 \leq i \leq k$, and for each query $q \in Q_i$, there is a vertex $v(P_i, q)$. Thus,

$$V_1 = \{v(P_i, q) \mid 1 \leq i \leq k \text{ and } q \in Q_i\}$$

Vertex $v(P_i, q)$ can only be assigned to the labels corresponding to $(P_i, q_i)$, i.e.,

$$L(v(P_i, q)) = \{\ell(P_i, q, A) \mid A \in A(q)\}$$

Note that assigning $v(P_i, q)$ to a label in $L(v(P_i, q))$ defines an answer of prover $P_i$ to query $q$.

**Constraint Vertices:** for each random string $r$, there is a vertex $v(r)$, i.e.,

$$V_2 = \{v(r) \mid r \in R\}$$

Vertex $v(r)$ can be assigned only to labels corresponding to $r$, i.e., $L(v(r))$ consists of labels $\ell(r, A_i, \ldots, A_k)$, such that $\forall i$, $A_i \in A(q_i(r))$ and $(A_1, \ldots A_k)$ are strongly consistent.

The edges of the graph are as follows. Every constraint vertex $v(r)$ is connected to every assignment vertex $v(P_i, q_i(r))$ by a unit-weight edge (see Figure 4.2).

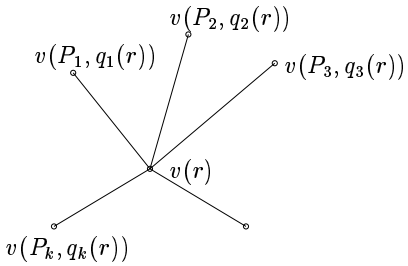


FIG. 4.2. *Edges incident to* $v(r)$

The graph is therefore a collection of stars that can have common leaves.

### 4.3. Hardness of Approximation Proof.

**4.3.1. Yes-Instances.** Assume that the input 3SAT(5) formula $\phi$ is a yes-instance. Consider a strategy of the provers for which the acceptance probability of the verifier is 1. For every prover $P_i$, $1 \leq i \leq k$, for every query $q \in Q_i$, let $f(q) \in A(q)$ be the answer of prover $P_i$ to query $q$ under this strategy. Note that for each random string $r$, $f(q_1(r)), \ldots, f(q_k(r))$ are strongly consistent. We define the following labeling of the graph $G$ (see Figure 4.3).

- For each random string $r \in R$, vertex $v(r)$ is assigned to label $\ell(r, f(q_1(r)), \ldots, f(q_k(r)))$.
- For each $i : 1 \leq i \leq k$, $q \in Q_i$, vertex $v(P_i, q)$ is assigned to label $\ell(P_i, q, f(q))$.

Consider an edge in the graph $G$ between $v(r)$ and $v(P_i, q_i(r))$, $r \in R$, $1 \leq i \leq k$. Vertex $v(r)$ is assigned to label $\ell(r, f(q_1(r)), \ldots, f(q_k(r)))$ and vertex $v(P_i, q_i(r))$ is assigned to label $\ell((P_i, q_i(r), f(q_i(r)))$. Thus, the separation cost of the edge is $\frac{1}{2}$, since the distance between the two labels is $\frac{1}{2}$. Hence, the total cost of the solution is $\frac{1}{2} \cdot k \cdot |R|$.

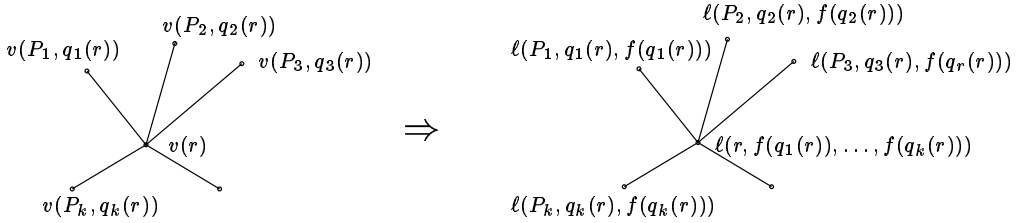


FIG. 4.3. *Yes instance: the embedding of edges incident to $v(r)$.*

**4.3.2. No-Instances.** Assume that the input 3SAT(5) formula $\phi$ is a no-instance. We prove that the cost of any solution to the metric labeling instance is at least $\binom{k}{2} \cdot \frac{\epsilon}{3} \cdot |R|$, and thus the gap between the Yes and the No instances is $\Omega(k)$. Observe that the assignment of the query vertices to query labels defines a strategy of the provers. We concentrate on this strategy and define the set $T \subseteq R \times [k] \times [k]$.

DEFINITION 4.3. *For $r \in R$, $1 \leq i < j \leq k$, $(r, i, j) \in T$ if and only if the answers of provers $P_i$ and $P_j$ to queries $q_i(r)$ and $q_j(r)$, respectively, are not weakly consistent (under the above strategy).* The following proposition is a direct consequence of Theorem 4.2.

PROPOSITION 4.4. $|T| \geq \binom{k}{2} \cdot \frac{\epsilon}{3} \cdot |R|$.

Consider an edge $e \in E$ and assume that the endpoints of the edge are assigned to labels $\ell_1$ and $\ell_2$. We denote by $\mathcal{P}_e$ the shortest path between the labels $\ell_1$ and $\ell_2$ in the graph of labels $G_L$. Note that the length of $\mathcal{P}_e$ is exactly the cost paid by edge $e$, and the solution cost is $\sum_{e \in E} |\mathcal{P}_e|$. We define the set $T' \subseteq R \times [k] \times [k]$ as follows. Consider a random string $r \in R$ and a pair of provers $P_i$ and $P_j$, $1 \leq i, j \leq k$, $i \neq j$. Let $e$ be the edge between $v(r)$ and $v(P_i, q_i(r))$. Then, $(r, i, j) \in T'$ if and only if the path $\mathcal{P}_e$ contains a label belonging to prover $P_j$ (i.e., a label of the form $\ell(P_j, q, A)$, for some $q \in Q_j, A \in A(q)$). Observe that the cost of the solution is at least $|T'|$.

LEMMA 4.5. *For $r \in R$, suppose $(r, i, j) \in T$, where $1 \leq i < j \leq k$. Then, either $(r, i, j) \in T'$, or $(r, j, i) \in T'$.*

*Proof.* Suppose that vertex $v(r)$ is assigned to label $\ell(r, A_1, \ldots, A_k)$, and suppose vertices $v(P_i, q_i(r))$ and $v(P_j, q_j(r))$ are assigned to labels $\ell(P_i, q_i(r), A'_i)$ and $\ell(P_j, q_j(r), A'_j)$, respectively. As $(r, i, j) \in T$, the answers $A'_i$ and $A'_j$ of provers $P_i$ and $P_j$ cannot be weakly consistent. However, the answers $A_i$ and $A_j$ are strongly

consistent. Therefore, either the $(i,j)$ coordinates in $A_i$ and $A'_i$ differ (recall that this coordinate contains an assignment to a clause $C_{ij}$), or the $(i,j)$ coordinates in $A_j$ and $A'_j$ differ (this coordinate contains an assignment to a distinguished variable $x_{ij}$). Assume the former is true (the other case is handled similarly).

Let $e$ be the edge between $v(r)$ and $v(P_i, q_i(r))$. It is enough to show that the path $\mathcal{P}_e$ contains a label corresponding to prover $P_j$. Suppose this is not the case. Let $\ell(P_a, q_a, A)$ and $\ell(P_b, q_b, A')$ be two consecutive query labels on the path. As the two labels are at distance 1, there must be an $r' \in R$, such that $q_a = q_a(r')$ and $q_b = q_b(r')$, and the answers $A$ and $A'$ are strongly consistent. As $a, b \neq j$, the $(i,j)$ coordinate in $q_a$ and in $q_b$ must contain some clause, and the two clauses are identical. Moreover, coordinate $(i,j)$ of $A$ and $A'$ must contain an identical assignment to the variables of this clause. Therefore, if path $\mathcal{P}_e$ starts at $\ell(P_i, q_i(r), A'_i)$, and does not pass through any label belonging to prover $P_j$, then for every query label $\ell(P_s, q_s, A)$ appearing on the path, coordinate $(i,j)$ of $q_s$ contains the same clause as that of $q_i(r)$, and coordinates $(i,j)$ in $A$ and $A'_i$ are identical. This is also true for the last query label on the path, denoted by $\ell(P_d, q_d, A_d)$. But this label is connected by an edge to label $\ell(r, A_1, \ldots, A_k)$, and therefore coordinates $(i,j)$ of $A_d$ and $A_i$ must be identical, which is impossible. □

It follows from the lemma that $|T'| \geq |T|$, yielding that the solution cost is at least $\binom{k}{2} \cdot \frac{\epsilon}{3} \cdot |R|$.

**4.3.3. The Hardness Factor.** The gap between the cost of the Yes-Instance and the No-Instance solutions is $\Omega(k)$. The size of the construction is dominated by the number of labels. For each $i$, $1 \leq i \leq k$, $|Q_i| \leq (5n)^{k^2}$, and for each $q \in Q_i$, $|A(q)| \leq 7^{k^2}$, and therefore the number of query labels is at most $k(5n)^{k^2} \cdot 7^{k^2}$. The size of $R$ is at most $(5n)^{k^2}$ and for each $r \in R$ the number of $k$-tuples of consistent answers is at most $7^{k^2}$. Hence, the number of constraint labels is bounded by $(5n)^{k^2} \cdot 7^{k^2}$. The construction size is therefore $N = n^{O(k^2)}$. If $k$ is a constant, then it is polynomial in $n$. Choosing $k = \text{poly}(\log n)$, we get that $k = (\log N)^{\frac{1}{2}-\delta}$ for arbitrarily small constant $\delta > 0$.

Thus, we have proved the following result.

THEOREM 4.6. *There is no efficient constant factor approximation algorithm for the metric labeling problem, unless P=NP. Moreover, for any constant* $0 < \delta < 1/2$, *there is no* $\Omega((\log N)^{\frac{1}{2}-\delta})$*-approximation algorithm for the problem, unless NP* $\subseteq$ $DTIME(n^{poly(\log n)})$.

**Acknowledgements.** The authors would like to thank Sanjeev Khanna for helpful comments on the presentation of the paper.

REFERENCES

[1] A. ARCHER J. FAKCHAROENPHOL, C. HARRELSON, R. KRAUTHGAMER, K. TALWAR AND E. TARDOS, *Approximate Classification via Earthmover Metrics*, in Proceedings of the 15th Annual ACM-SIAM Symposium on Discrete Algorithms, New Orleans, LA, 2005, SIAM, Philadelphia, 2005, pp. 1079–1087.

[2] S. ARORA, C. LUND, R. MOTWANI, M. SUDAN, AND M. SZEGEDY, *Proof verification and the hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555.

[3] S. ARORA AND S. SAFRA, *Probabilistic checking of proofs: A new characterization of* NP, J. ACM, 45 (1998), pp. 70–122.

[4] Y. BARTAL, *Probabilistic approximation of metric spaces and its algorithmic applications*, in Proceedings of the 37th IEEE Symposium on Foundations of Computer Science, Burlington, VT, 1996, IEEE Press, Piscataway, NJ, pp. 184–193.

[5] Y. Bartal, *On approximating arbitrary metrics by tree metrics*, in Proceedings of the 30th Annual ACM Symposium on Theory of Computing, Dallas, TX, 1998, ACM, New York, 1998, pp. 161–168.

[6] G. Călinescu, H. Karloff, and Y. Rabani, *An improved approximation algorithm for multiway cut*, J. Comput. System Sci., 60 (2000), pp. 564–574.

[7] G. Călinescu, H. Karloff, and Y. Rabani, *Approximation algorithms for the 0-extension problem*, SIAM J. Comput., 34, 2005, pp. 358–372.

[8] C. Chekuri, S. Khanna, J. Naor, and L. Zosin, *Approximation algorithms for the metric labeling problem via a new linear programming formulation*, SIAM J. Discrete Math., 18 (2004), pp. 608–625.

[9] E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis, *The complexity of multiterminal cuts*, SIAM J. Comput., 23, 1994, pp. 864–894.

[10] J. Fakcharoenphol, C. Harrelson, S. Rao and K. Talwar, *An improved approximation algorithm for the 0-extension problem*, in Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms, Baltimore, MD, 2003, SIAM, Philadelphia, 2004, pp. 342-352.

[11] J. Fakcharoenphol, S. Rao, and K. Talwar, *A tight bound on approximating arbitrary metrics by tree metrics*, in Proceedings of the 35th Annual ACM Symposium on Theory of Computing, San Diego, CA, 2003, ACM, New York, 2003, pp. 448–455.

[12] U. Feige, *A Threshold of ln n for approximating set cover*, J. ACM, 45 (1998), pp. 634–652.

[13] A. Gupta and E. Tardos, *A Constant Factor Approximation Algorithm for a Class of Classification Problems*, in Proceedings of the ACM Symposium on the Theory of Computing, Portland, OR, 2000, ACM, New York, 2000, pp. 652–658.

[14] D. Karger, P. Klein, C. Stein, M. Thorup, and N. Young, *Rounding algorithms for a geometric embedding of minimum multiway cut*, in Proceedings of the 31st Annual ACM Symposium on the Theory of Computing, Atlanta, GA, 1999, ACM, New York, 1999, pp. 668–678.

[15] H. Karloff, S. Khot, A. Mehta and Y. Rabani, *On earthmover distance, metric labeling, and 0-extension*, in Proceedings of the 38th Annual ACM symposium on Theory of Computing, Seattle, WA, 2006, ACM, New York, 2006, pp. 547–556.

[16] A. Karzanov, *Minimum 0-extension of graph metrics*, Europ. J. Combinat., 19 (1998), pp. 71–101.

[17] A. Karzanov, *A combinatorial algorithm for the minimum* $(2, r)$*-metric problem and some generalizations*, Combinatorica, 18 (1999), pp. 549–569.

[18] J. Kleinberg and E. Tardos, *Approximation algorithms for classification problems with pairwise relationships: metric labeling and markov random fields*, J. ACM, 49 (2002), pp. 616–630.

[19] R. Raz, *A parallel repetition theorem*, SIAM J. Comput., 27, 1998, pp. 763–803.