Metric Labeling: Upper and Lower Bounds

SEFFI NAOR

COMPUTER SCIENCE DEPT.
TECHNION
HAIFA, ISRAEL

Based on Joint Work with:
CHANDRA CHEKURI, JULIA CHUZHOY, SANJEEV KHANNA, ROY SCHWARTZ, AND LEONID ZOSIN.

Metric Labeling: The Problem

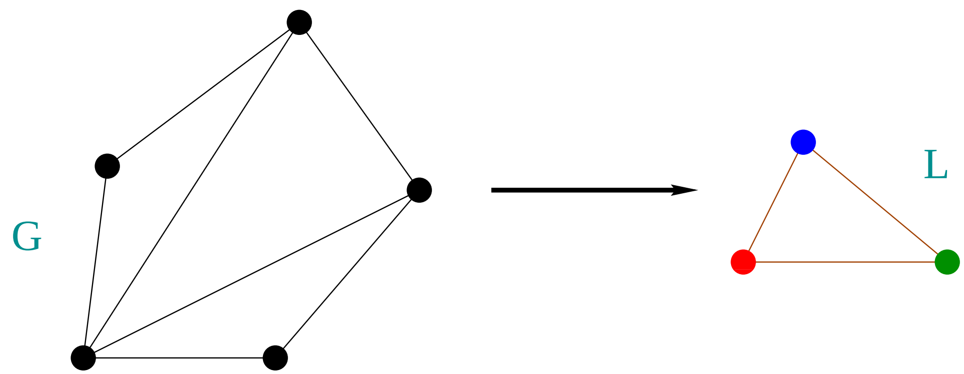
- **Input:**
  - Undirected graph G with edge weights $w(u,v)$.
  - A set L of k labels equipped with a metric d.
  - Cost function $c : V(G) \times L \to \mathcal{R}$.

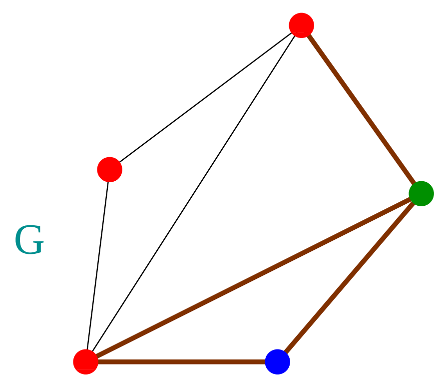
- **Goal:** An assignment $f : V(G) \to L$ (or a labeling of $V(G)$).

- **Objective Function:** minimize

$$\underbrace{\sum_{u \in V(G)} c(u, f(u))}_{\text{Labeling Cost}} + \underbrace{\sum_{u,v} w(u,v) d(f(u), f(v))}_{\text{Separation Cost}}$$

Example





Combinatorial Optimization: Related Problems

- **Multiway Cut:**
  - Set of terminals $t_1, \ldots, t_k$.
  - Find minimum cut separating the terminals.
  - Special case of ML: uniform metric and no assignment cost.

- **0-Extension:**
  - Same as multiway cut except that metric is arbitrary:
    penalty of cut edge depends on terminals that endpoints belong to.
  - Special case of ML.

- **Quadratic Assignment:** dropping the bijective property in QA yields metric labeling.

Motivation

- Clean and general abstraction of classification problems [Kleinberg and Tardos, 1999].
- Links to Markov random fields and their applications.
- Specific applications to image processing and analysis.
- Generalization of well known optimization problems.

Do assignment costs matter?

The $(0, \infty)$-Extension Problem:

$$c(u, i) \in \{0, \infty\} \text{ for all } u \in V(G), 1 \leq i \leq k.$$

- Approximation preserving reduction from metric labeling with arbitrary assignment costs to $(0, \infty)$-extension.

- Reduction preserves label set, but changes graph (in a simple way).

Theorem. [Chuzhoy 2001] *If there is a $f(n, k)$-approximation algorithm for $(0, \infty)$-extension, then there is a $f(n+nk, k)$-approximation algorithm for general metric labeling.*

Relaxation: Embedding in a Simplex

[Chekuri, Khanna, N., Zosin, 2001]

- For each $v \in V$: $v \longmapsto (x(v,1), x(v,2), \ldots x(v,k))$, where

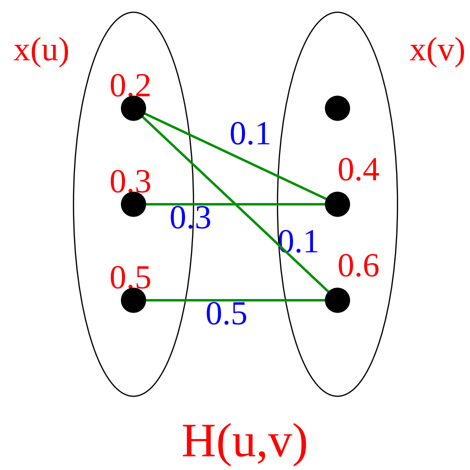
$$\sum_{i=1}^{k} x(v,i) = 1$$

Vertex v is mapped into a probability distribution over the label set.

- Distance between u and v defined by Earthmover Metric - solution to a transportation problem between $(u, 1), \ldots (u, k)$ and $(v, 1), \ldots, (v, k)$ with respect to label metric d.

$$d_{EM}(u, v) = \sum_{i,j} d(i, j) \cdot x(u, i, v, j)$$

$x(u, i, v, j)$ - flow on edge $((u, i), (v, j))$



Linear Program: Computing the Embedding

- **Result:** Embedding in a simplex where distances are defined by an earthmover metric (and not ℓ_1).
- **Objective Function:** Minimize

$$\underbrace{\sum_{u \in V} \sum_{i=1}^{k} c(u,i) \cdot x(u,i)}_{\text{labeling cost}} + \underbrace{\sum_{(u,v) \in E} w(u,v) \sum_{1 \le i,j \le k} d(i,j) \cdot x(u,i,v,j)}_{\text{separation cost}}$$

Constraints

$$
\begin{aligned}
\sum_{i=1}^{k} x(u,i) &= 1 \quad \forall\, u \in V \\
\sum_{j=1}^{k} x(u,i,v,j) - x(u,i) &= 0 \qquad \forall\, u,v \in V, i \in 1,\ldots,k \\
x(u,i,v,j) - x(v,j,u,i) &= 0 \qquad \forall\, u,v \in V,\, i,j \in 1,\ldots,k \\
x(u,i), x(u,i,v,j) &\geq 0
\end{aligned}
$$

Uniform Metric

- For any $i \neq j$, $d(i, j) = 1$.
- What does the earthmover solution look like? for edge (u, v):

$$x(u, i, v, i) = \min\{x(u, i), x(v, i)\}$$

- Thus,

$$d_{EM}(u, v) \; = \; \sum_{i,j} d(i, j) x(u, i, v, j) \; \geq \; \frac{1}{2} \cdot \sum_{i=1}^{k} |x(u, i) - x(v, i)|$$

Uniform Metric: Rounding Algorithm

Rounding an LP solution. [Kleinberg and Tardos, 1999].

Idea: Random choices should be correlated.

Algorithm: repeat until all vertices are labeled.

1. pick i at random from $\{1, 2, \ldots, k\}$.
2. pick θ at random from the interval $[0, 1]$.
3. label an unlabeled vertex u with i iff $\theta \leq x(u, i)$.

Uniform Metric: Integrality Gap

Observation: Probability of assigning i to u is exactly $x(u, i)$.

Lemma: Probability that u and v get different labels is at most

$$\sum_{i=1}^{k} |x(u,i) - x(v,i)|$$

Recall: $d_{EM}(u, v) \;\geq\; \frac{1}{2} \cdot \sum_{i=1}^{k} |x(u,i) - x(v,i)|$

Theorem: For a uniform metric, integrality gap ≤ 2.

Open Question: Can the 2-approximation be improved?

General Metrics

- Solve the simplex embedding LP.
- Approximate the fractional solution to the LP by a deterministic HST metric losing a factor of $O(\log k)$.
- The integrality gap on an HST tree is $O(1)$.
- Yielding an $O(\log k)$-approximation for general metrics [Kleinberg and Tardos, 1999].

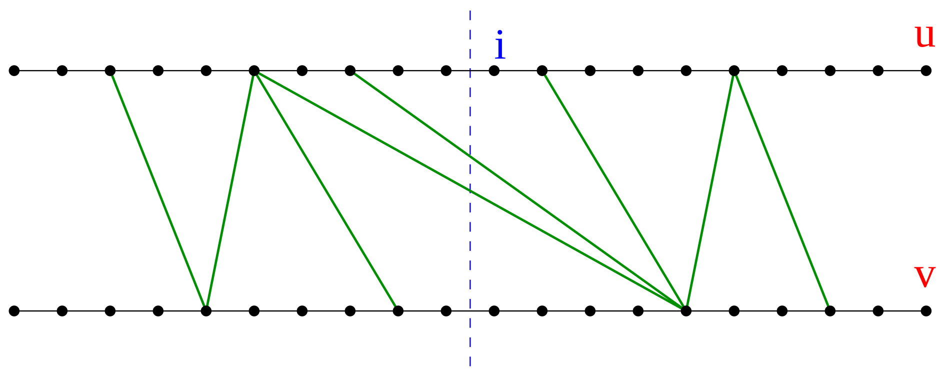
Linear Metric

Rounding of LP solution:

- Assume w.l.o.g. labels are integers $1, 2, \ldots, k$.
- For each vertex u, define $\alpha(u, i) = \sum_{j=1}^{i} x(u, j)$.
- Pick θ uniformly at random from $[0, 1]$.
- $L(u) = i$ iff $\alpha(u, i-1) < \theta \leq \alpha(u, i)$.
- All vertices get a label since $\alpha(u, k) = 1$.

Lemma 1: $$d_{EM}(u,v) \geq \sum_{i=1}^{k} |\alpha(u,i) - \alpha(v,i)|.$$

Flow is uncrossing



Flow crossing i is exactly $|\alpha(u,i) - \alpha(v,i)|$. $\square$

Analysis

Lemma 1: $$d_{EM}(u,v) \geq \sum_{i=1}^{k} |\alpha(u,i) - \alpha(v,i)|.$$

Lemma 2: $$\mathbf{E}\left[d((L(u), L(v))\right] = \sum_{i=1}^{k} |\alpha(u,i) - \alpha(v,i)|.$$

$$\Downarrow$$

Theorem: The integrality gap of the LP for the line metric is 1.

Convex functions on the line

- $d(i,j) = f(|i-j|)$ where f is convex and increasing.
- d is a metric iff f is *linear*.
- The linear programming formulation is useful for convex f.
- Integrality gap is 1 since flow is uncrossing.

Truncated Linear Metric

- $d(i, j) = \min\{M, |i - j|\}$.
- Applications to image processing.
- Generalizes uniform and linear metrics and is NP-hard.
- $2 + \sqrt{2} \simeq 3.414$-approximation by generalizing the linear algorithm. [Chekuri, Khanna, N., Zosin, 2001]
- **Open Question**: Improve the approximation factor.

Truncated Quadratic Distance

- $d(i,j) = \min\{(i-j)^2, M\}$. Not a metric!
- Useful function for vision applications.
- $O(\sqrt{M})$-approximation easy.
- **Open Questions:**
  - NP-hard?
  - LP gap?
  - $O(1)$ approximation?

0-Extension Problem

- **Input:**
  - Graph G with edge weights $w(u, v)$.
  - $T \subset V(G)$ - Set of k terminals.
  - d - Metric on T.

- **Solution**: Partitioning of the graph, s.t. each terminal is in a different connected component.
  - $t(v)$ - terminal in connected component of v.

- **Objective**: minimize $\displaystyle\sum_{(u,v)\in E(G)} w(u, v) \cdot d(t(u), t(v))$.

0-Extension Problem: Open Questions

- Is 0-extension easier than $(0, \infty)$-extension?

- I.e., if each non-terminal vertex can be labeled for free, does that make the metric labeling problem easier?

- Best approximation factor known: $O\left(\frac{\log k}{\log \log k}\right)$ [FHRT] for general metrics (improving a previous factor of $O(\log k)$ [CKR]).

Balanced Metric Labeling

- **Input:** Metric labeling instance.

- **Additional constraint:**

  Each label can be assigned to at most ℓ vertices.

  [N., Schwartz, STOC 2005]

Motivation

- Minimum weight k-way balanced partitioning:
  - Each part contains at most $2n/k$ vertices.
  - Minimizing weight of edge cuts.
- Special case of balanced metric labeling:
  - Label is equivalent to a Part.
  - $\ell \leq 2n/k$.
  - Uniform metric.

Motivation (contd.)

- What if each vertex can only be labeled by a subset of the labels?
  - The balanced $\{0, \infty\}$-extension problem.
- Application: Clustering Base Transceiver Stations in GSM networks:
  - Weighted graph on the BTS-s: traffic $\mapsto$ edge weight.
  - Each cluster is controlled by a Base Station Controller (= label).
  - Base Station Controller have bounded capacity.
  - Each BTS can only be assigned to a subset of the BSC-s.
- Graph arrangement problems:
  - E.g., linear-arrangement: linear metric and capacity = 1.

Balanced Uniform Metric Labeling - Difficulties

- Bounding the number of vertices assigned to each label?
  - Not obvious in the methods developed for uncapacitated uniform metric labeling, e.g., the Kleinberg-Tardos algorithm.
- Incorporating label assignment costs?
  - Not obvious in the techniques developed for approximating graph partitioning problems ([LR], [ENRS], and [ARV]).
  - For example, there may not always exist a label that can be assigned to all vertices in a single cluster of the partition.

Spreading Constraints

- Very useful for approximating graph partitioning problems.
- Example: $\forall S \subseteq V, \forall u \in S$: $\displaystyle\sum_{v \in S} d(u,v) \geq |S| - \ell$.
- For large subsets S, there is a radius guarantee:

$$\exists v \in S: \; d(u,v) \geq 1 - \frac{\ell}{|S|}$$

- Radius guarantee $\Rightarrow$ Ball growing techniques can be applied.

The Relaxation

- Embedding in a k-dimensional simplex.
- Spreading constraints.
- Capacity constraints:

$$\forall \text{ label } j : \qquad \sum_{v \in V} x(v, j) \leq \ell$$

- Closeness constraints.

The Relaxation: Closeness Constraints

- Closeness of u and v wrt label j: $c_j(u,v) \le x(u,j), x(v,j)$.
- Variation distance: $\forall u, v$,

$$d(u,v) = 1 - \sum_{j \in L} c_j(u,v)$$

- Triangle inequality: $\forall u, v, w \in V$,

$$\sum_j \left| c_j(u,v) - c_j(u,w) \right| \le 1 - \sum_j c_j(v,w)$$

The Approximation Algorithm

- **Overview:** A combination of randomized metric decomposition and label assignment techniques.

- **Initial Labeling:** Each vertex v is assigned a root labeling, $f^* : V \rightarrow L$, satisfying:

$$\mathbb{P}\mathrm{r}[f^*(v) = j] = x(v, j) \quad , \quad \forall v \in V, \forall \text{ label } j.$$

- **Iteratively:** Each vertex, in its turn, is a root and labels a subset of the unlabeled vertices.

Radius and Label Tests

- **Current root:** Vertex u.
- **Radius test:**
  - Choose radius R from the distribution:

$$f_R(r) = \left(\frac{n}{n-1}\right) \cdot \frac{1+\varepsilon}{\varepsilon} \cdot \ln n \cdot n^{-r \cdot \frac{1+\varepsilon}{\varepsilon}}, \quad r \in \left[0, \frac{\varepsilon}{1+\varepsilon}\right]$$

  - Define a ball of radius R, with respect to metric d, around root vertex u:

$$\{x \mid d(u,x) \leq R\}$$

Radius and Label Tests (contd.)

- **Label Test:**
  - Choose uniformly in random $\alpha \in [0, x(u, f^*(u))]$.
  - Define vertices close to the root u with respect to root label $f^*(u)$:

$$\{x \mid c_{f^*(u)}(u, x) \geq \alpha\}$$

- **Labeling:** All unlabeled vertices that pass both radius and label tests receive label $f^*(u))$.

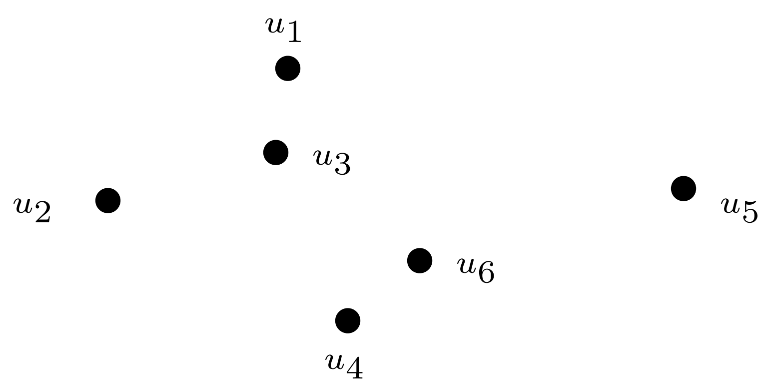
Approximation Algorithm: Summary

- For each $u \in V$, iteratively:
  - Apply radius and label test.
- Output labeling.

Theorem: Upon termination, all vertices are labeled.

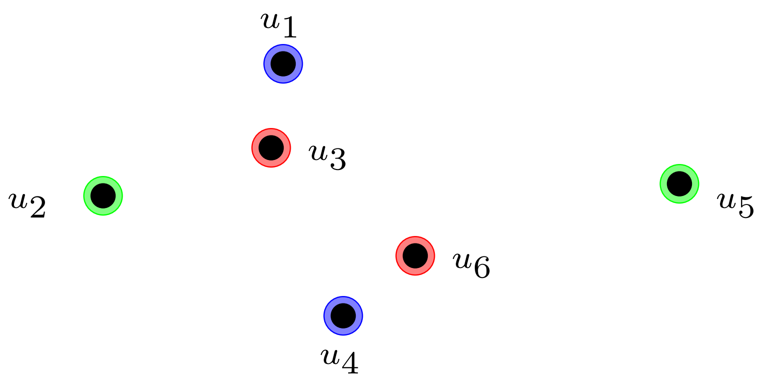
Proof: Each vertex passes the radius and label tests when it becomes the root vertex.

The Approximation Algorithm - Example



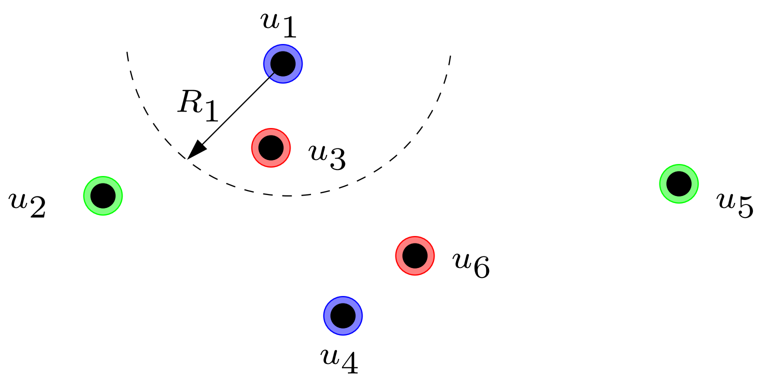
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



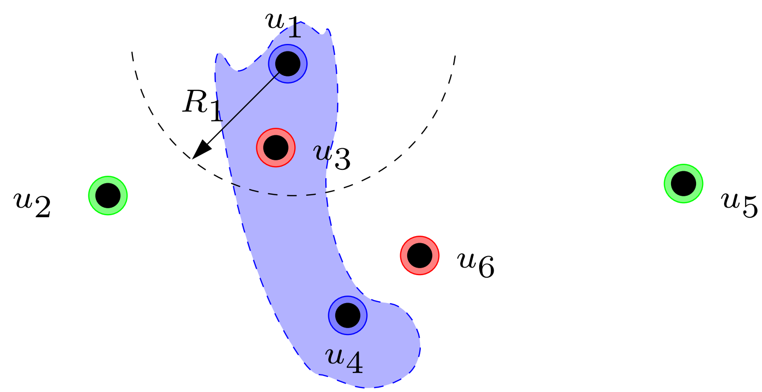
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



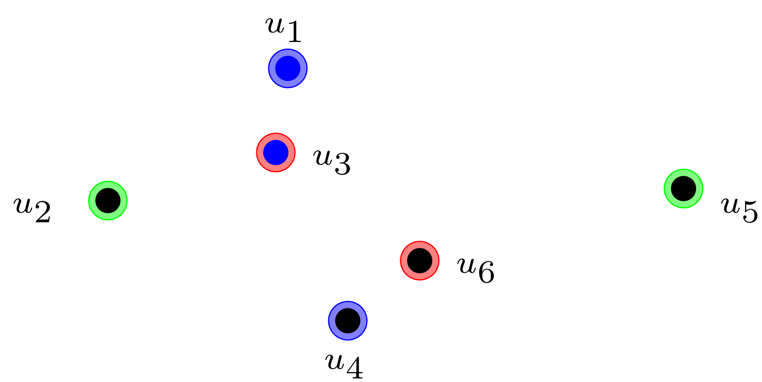
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



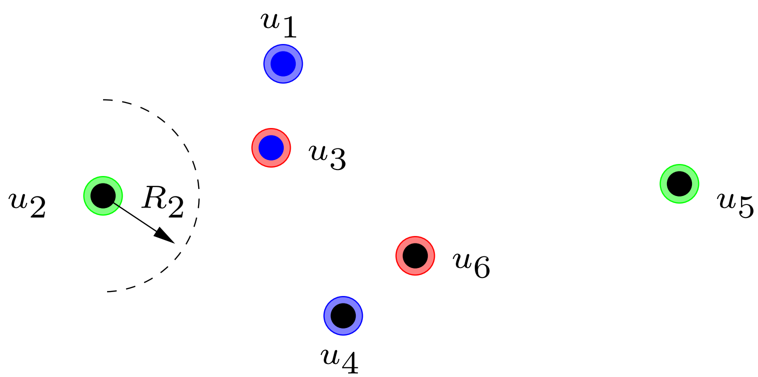
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



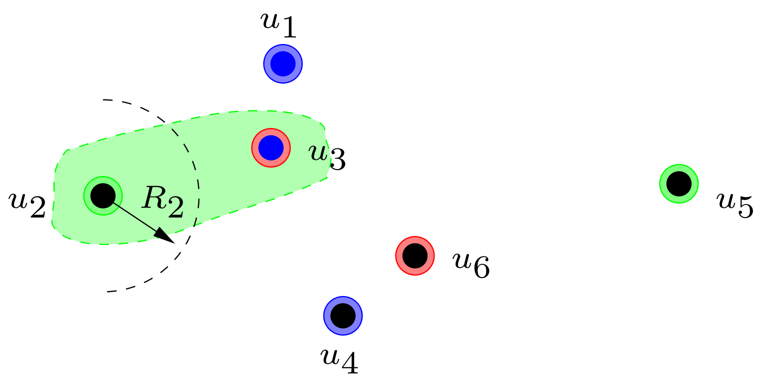
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



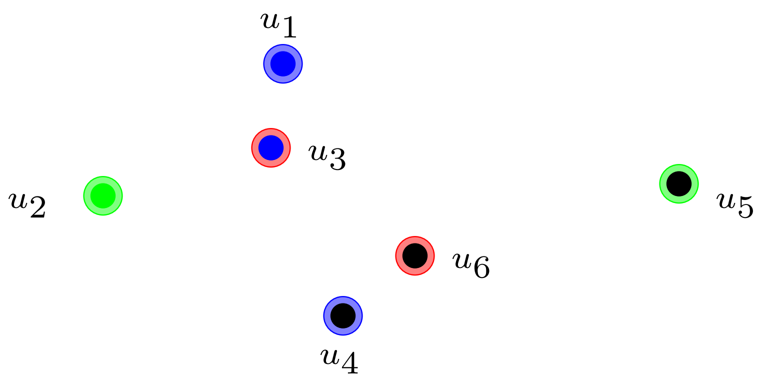
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



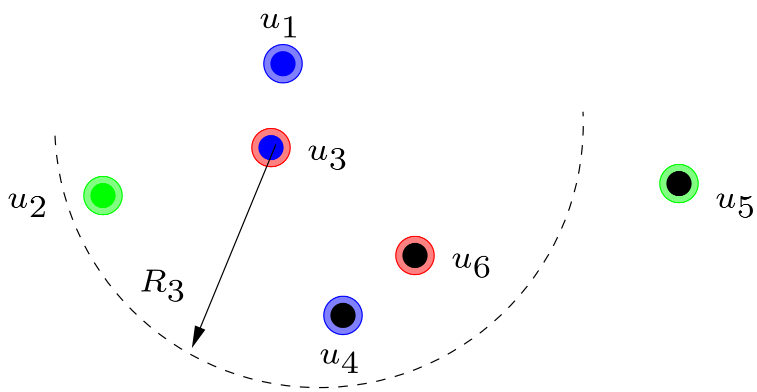
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



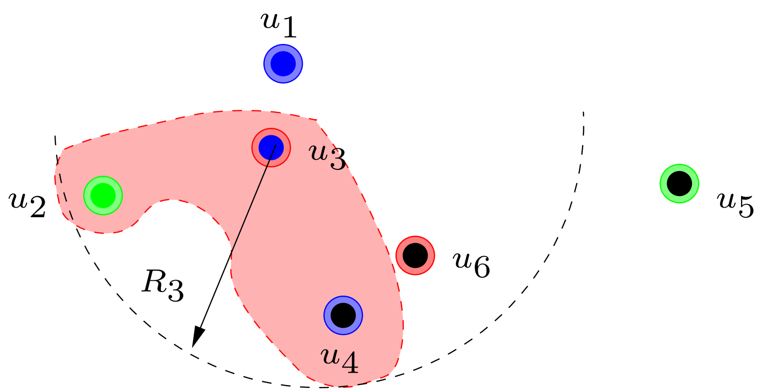
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



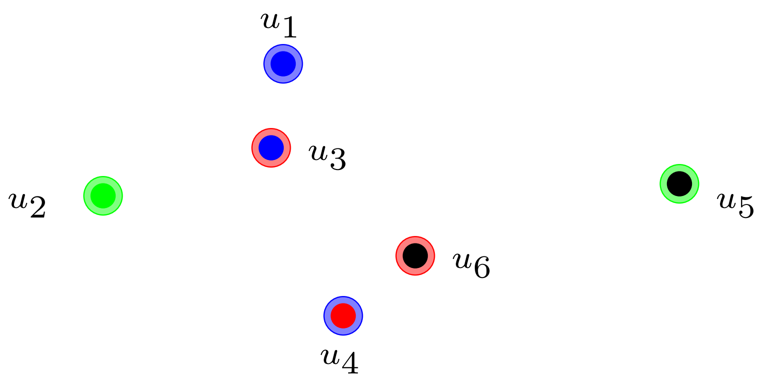
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



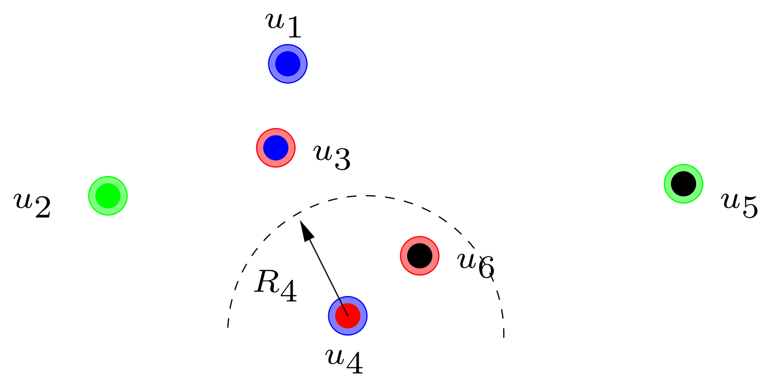
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



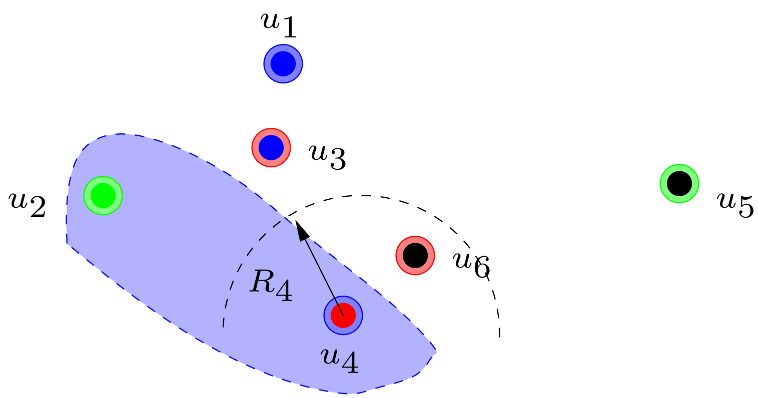
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



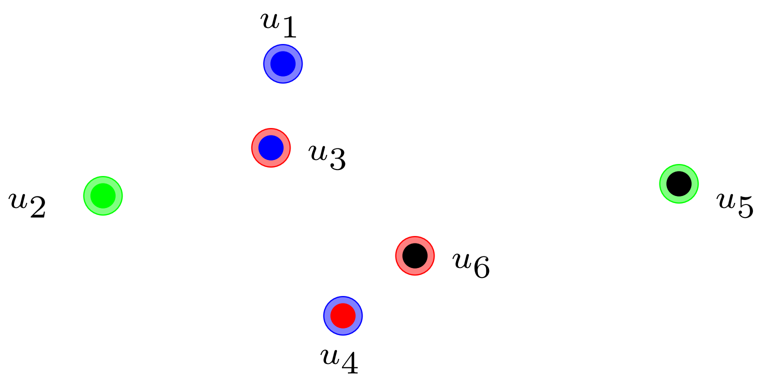
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



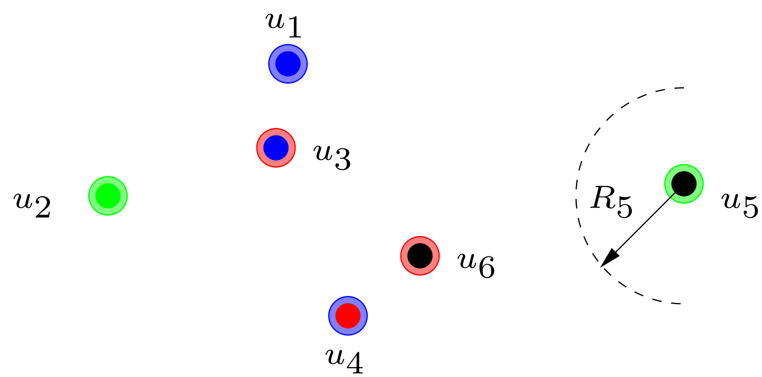
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



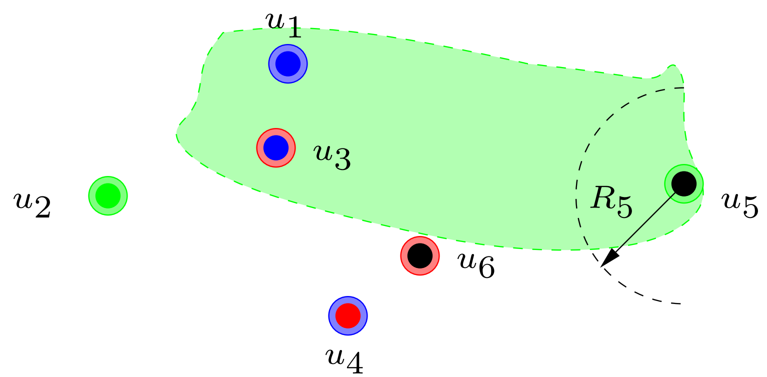
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



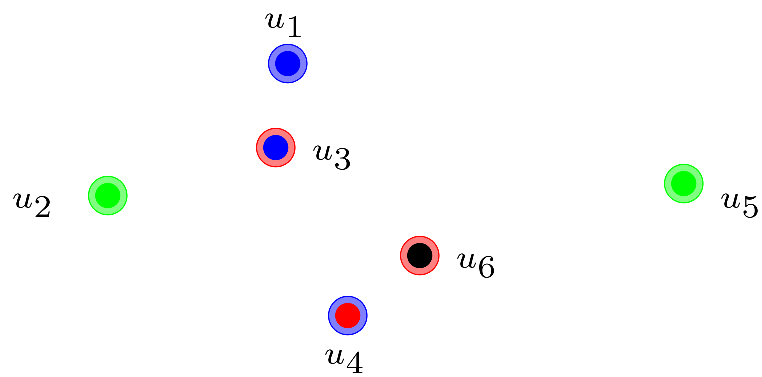
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



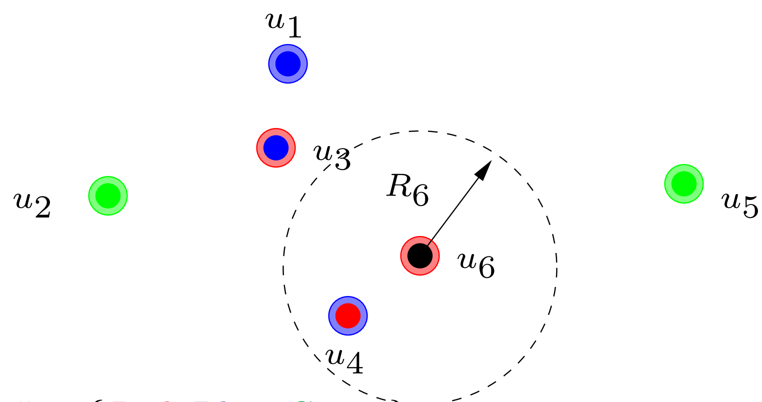
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



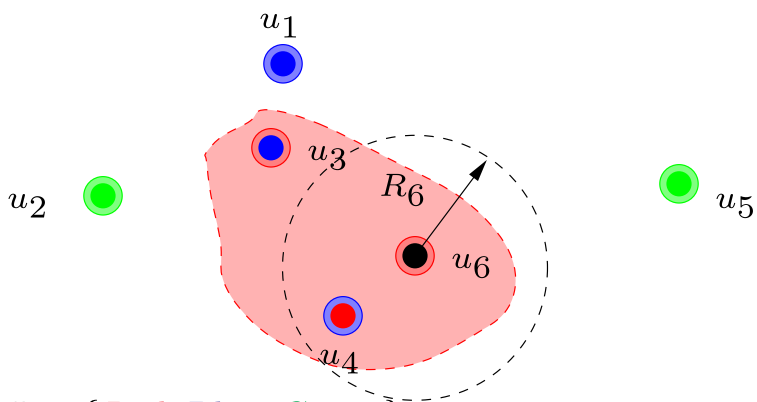
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



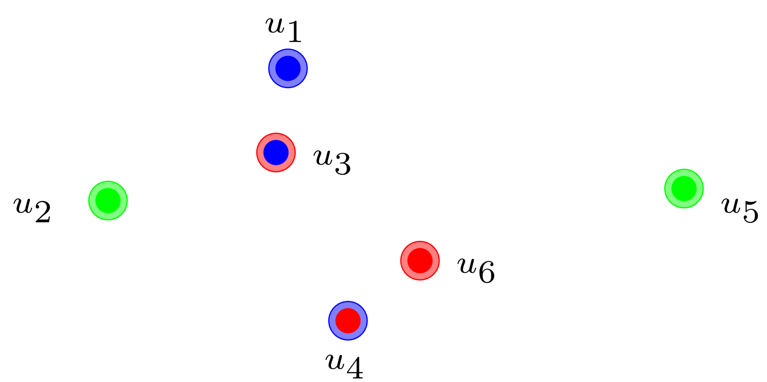
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



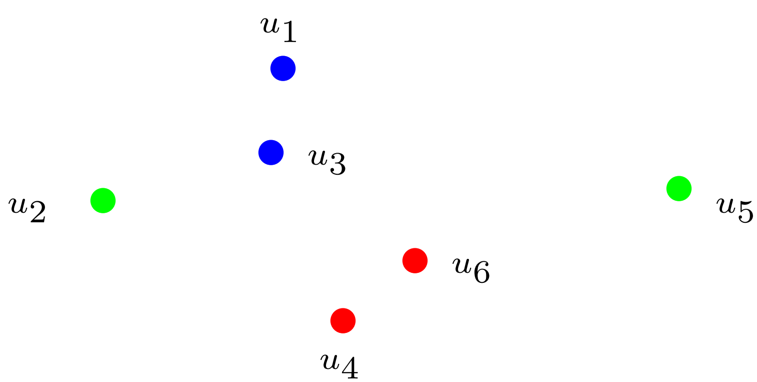
$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



$L = \{Red, Blue, Green\}$

The Approximation Algorithm - Example



$L = \{Red, Blue, Green\}$

Analysis

- **Difficulty:**
  - Capacity: Easy to bound the number of vertices assigned to a label with independent random labels.
  - Vertex separation costs: If the labels chosen for the vertices are dependent [KT], cost of vertex separation is bounded.

Analysis (contd.)

- **Main Ingredient:** The algorithm balances the dependencies between the labels assigned to the vertices.

  – Label of a vertex depends on only a limited number of other labels: Labels of vertices that are far from each other are independent.

  – Spreading constraints: not too many vertices are close.

  – Number of vertices assigned to each label is bounded via a new inequality of Janson for tail bounds of (partly) dependent random variables.

  – Separation cost is bounded.

Approximation Factor

- Bicriteria approximation factor: For any $0 < \varepsilon < 1$,
  - $O\left(\frac{\ln n}{\varepsilon}\right)$-approximation to the solution cost.
  - $\min\left\{\frac{O(\ln k)}{1-\varepsilon}, \ell + 1\right\}(1+\varepsilon)\,\ell$ vertices are assigned to each label.
- For $\ell = O(1)$ or $k = O(1)$, capacity is violated by a constant multiplicative deviation.
- Compare with balanced k-way partitioning:

  Either $(O(\log n), \mathrm{const})$, [ENRS] or $(O(\sqrt{\log n}\log k), \mathrm{const})$ [ARV].

Open Questions

- Can we improve the approximation factor?
- Can we obtain the same biciriteria factor $(\log n, \text{constant})$ known for balanced partitioning?

Hardness of Metric Labeling

- Back to uncapacitated metric labeling [Chuzhoy, N., FOCS 2004]:
- There is no constant approximation for Metric Labeling unless P=NP.
- No $\log^{\frac{1}{2}-\delta} n$-approximation exists unless NP $\subseteq$ DTIME$(n^{\text{poly} \log n})$ (for any constant δ).
- Hardness is proved for $(0, \infty)$-extension.

Gap 3SAT(5)

Input: A 3SAT(5) formula φ on n variables.

- φ is a YES-instance if it is satisfiable.
- φ is a NO-instance (with respect to some ε) if at most a $(1-\varepsilon)$-fraction of the clauses are simultaneously satisfiable.

Theorem: [ALMSS'92] There is some $0 < \varepsilon < 1$, such that it is NP-hard to distinguish between YES and NO instances.

A 2-prover Protocol for 3SAT(5) Formula φ

- Verifier: randomly chooses clause C and one of its variables x.
- Prover 1: receives the clause C and answers with an assignment to the variables of C that satisfy it.
- Prover 2: receives variable x and answers with an assignment to x.
- Verifier: checks that the two assignments match.

Theorem:
• If φ is a YES-instance: there is a strategy of the provers such that the verifier always accepts.
• If φ is a NO-instance: for any strategy, the acceptance probability is at most $\left(1 - \frac{\varepsilon}{3}\right)$.

The Raz Verifier

- Performs ℓ parallel repetitions of the 2-Prover Protocol.
- A query to prover 1 is an ℓ-tuple of clauses and a query to prover 2 is an ℓ-tuple of variables.
- If φ is a YES-instance: then there is a strategy of the two provers that makes the verifier always accept.
- If φ is a NO-instance: then for any strategy of the two provers the acceptance probability is at most $2^{-O(\ell)}$.

A Simple $(3-\varepsilon)$-Hardness

- Start from a 3SAT(5) formula φ.
- Use the Raz verifier with ℓ repetitions (ℓ is a large constant) to produce a $(0,\infty)$-extension instance:
  - If φ is a YES-instance, then there is a solution of cost $|R|$.
  - If φ is a NO-instance, then the cost of any solution is at least $(3-\delta)|R|$.

A $(3-\varepsilon)$-Hardness: Label Set

- $\forall$ query-answer pair (q,a) of each prover, there is a label $\ell(q,a)$.
- Given:
  - random string r.
  - queries q_1, q_2 sent to the provers under r.
  - a_1 and a_2 is a pair of consistent answers to q_1 and q_2.

  $\Longrightarrow$ There is an edge of length 1 between (q_1,a_1) and (q_2,a_2).
- Label distances are defined by shortest paths in the label graph.
- Label graph is bipartite: Part $\Leftrightarrow$ Prover. Distances: either 1, or ≥ 3.

A $(3-\varepsilon)$-Hardness: the Graph

- For each possible query q to provers 1 and 2 there is a vertex $v(q)$ that can only be assigned to its corresponding labels ($\ell(q,a)$).
- For each random string r, let q_1, q_2 be the queries sent to the two provers under r. There is an edge between $v(q_1)$ and $v(q_2)$.

Note that every assignment of the vertices to the labels defines a strategy for the provers and vice versa.

Properties

- If φ is a YES-instance:
  - $\exists$ strategy of provers s.t. their answers are always consistent.
  - Strategy defines an assignment of vertices to labels of cost $|R|$.
- If φ is a NO-instance:
  - Assignment of labels to vertices defines a strategy for the provers.
  - Acceptance probability of this strategy is at most $2^{-O(\ell)}$.
  - Hence, almost all the edges in the graph pay (at least) 3.
  - The solution cost is arbitrarily close to $3|R|$.

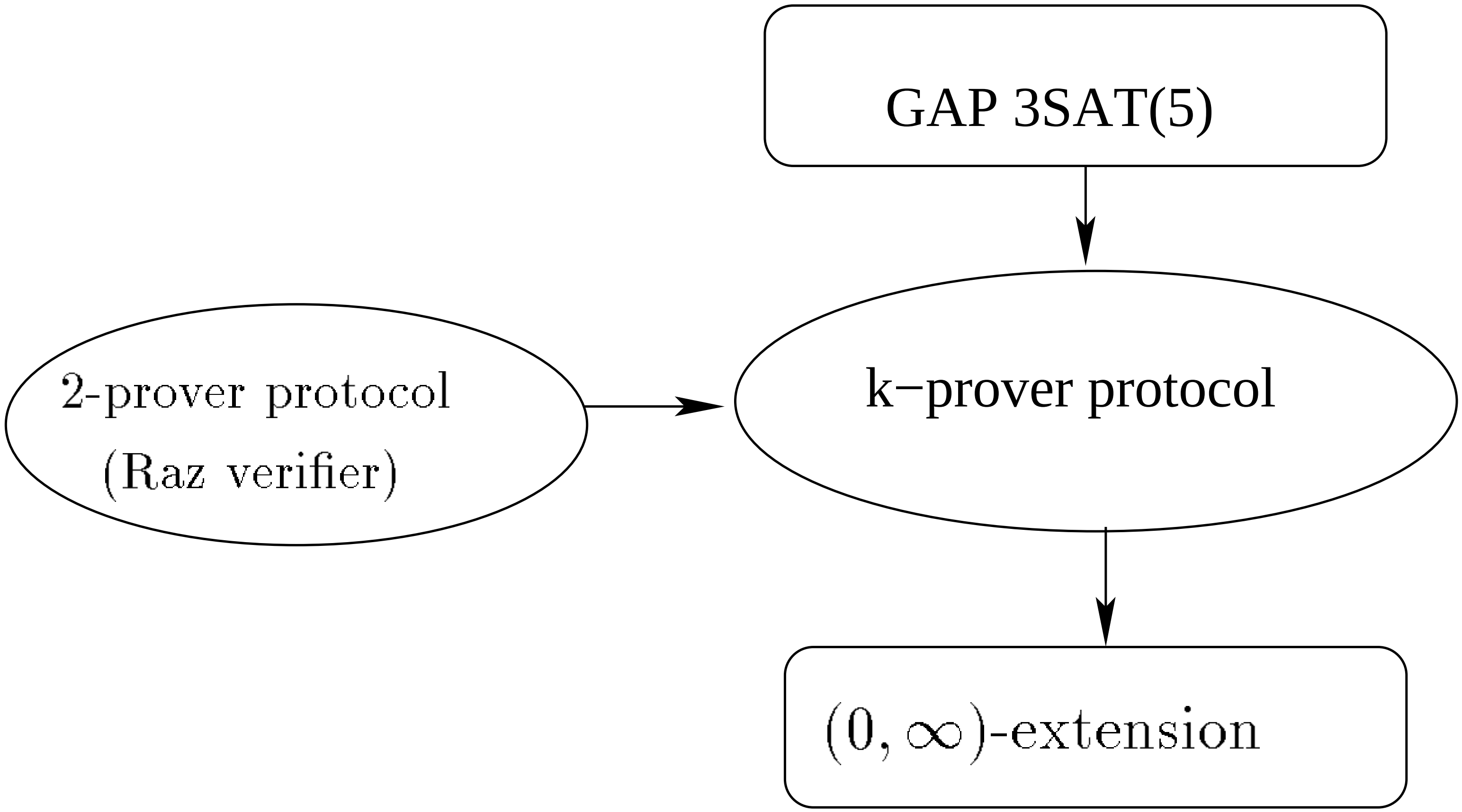
Extending to $\sqrt{\log n}$-Hardness

Difficulty:

- Suppose queries q_1 and q_2 are sent to the two provers.
- If their answers a_1,a_2 are inconsistent, then there is a path of length (precisely) 3 in the label graph between the labels $\ell(q_1, a_1)$ and $\ell(q_2, a_2)$.
- This is true even if the answers are inconsistent in many coordinates.

Goal: If the answers are inconsistent in many coordinates, the length of the path between them should also be large.

Plan


GAP 3SAT(5)
2-prover protocol
(Raz verifier)
k−prover protocol
$(0, \infty)$-extension

A New k-Prover System

For each pair of provers (i, j), $1 \leq i < j \leq k$:

- The verifier chooses randomly and independently clause C_{ij} and one of its variables x_{ij}.
- Prover i receives clause C_{ij} and answers with an assignment to its variables satisfying the clause.
- Prover j receives x_{ij} and answers with an assignment to it.
- Every other prover $a \neq i, j$ receives both C_{ij} and x_{ij} and answers with an assignment to the variables of C_{ij} satisfying the clause.

A Query

Each query has $\binom{k}{2}$ coordinates.

Coordinate (a, b) (for $a < b$) of the query for prover i:

- If $i = a$, it contains C_{ab}
- If $i = b$, it contains x_{ab}
- If $a, b \neq i$, it contains both C_{ab} and x_{ab}

Example: Queries in a 3-Prover Protocol

	$(1,2)$	$(1,3)$	$(2,3)$
P_1	$C_{1,2}$	$C_{1,3}$	$C_{2,3}, x_{2,3}$
P_2	$x_{1,2}$	$C_{1,3}, x_{1,3}$	$C_{2,3}$
P_3	$C_{1,2}, x_{1,2}$	$x_{1,3}$	$x_{2,3}$

The k-Prover System: Properties

Definition:

- Let A_i, A_j be the answers of provers i, j to their queries.
- The answers are weakly consistent if their (i, j) coordinates match.
- They are strongly consistent if all their coordinates match.

Theorem: If φ is a YES-instance, then there is some strategy of the provers, such that their answers are always strongly consistent.

Theorem: If φ is a NO-instance, then for every pair of provers, the probability that their answers are weakly consistent is at most $(1 - \frac{\varepsilon}{3})$.

The Reduction - an Overview

Given a 3SAT(5) formula φ on n variables, we use the k-prover system to produce an instance of $(0, \infty)$-extension, such that:

- If φ is a YES-instance, there is a solution of cost $\frac{k}{2}|R|$.
- If φ is a NO-instance, the cost of any solution is at least $|T| \geq \binom{k}{2}\frac{\varepsilon}{3}|R|)$
- Thus, the gap between YES and NO instances is $\Omega(k)$.
- The instance size is $N = n^{O(k^2)}$.

$\Rightarrow$ Choosing $k = \text{poly}(\log n)$, no $\log^{\frac{1}{2}-\delta} N$ approximation exists unless NP $\subseteq$ DTIME$(n^{\text{poly} \log n})$ (for any constant δ).

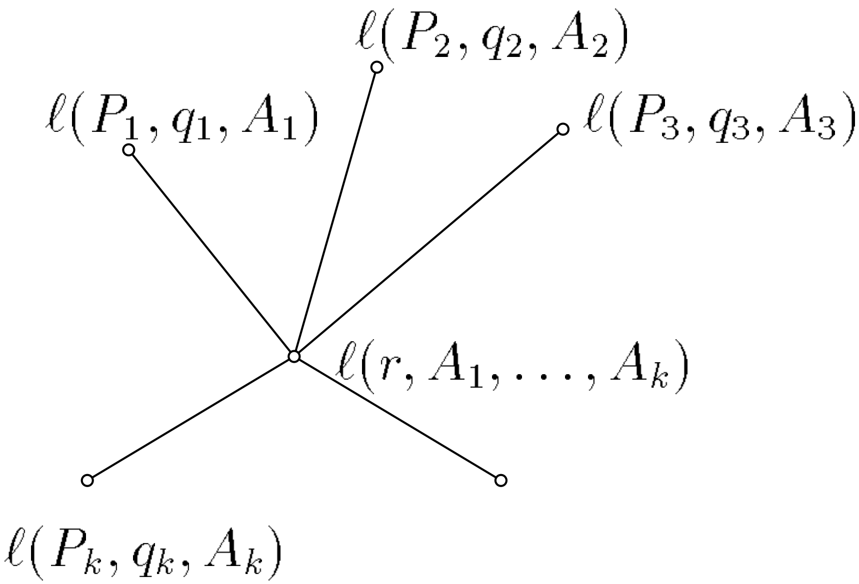
The Construction: Label Metric

There are two types of labels:

- Query Label $\ell(P_i, q_i, A_i)$:
  - For each prover P_i,
  - For each query q_i to prover P_i,
  - For each possible answer A_i to q_i.
- Constraint Label $\ell(r, A_1, \ldots, A_k)$:
  - For each random string r,
  - For each k-tuple $A_1, \ldots, A_k$ of strongly consistent answers of the provers to the queries implied by r.

Label Metric: Edges

Let r be a random string, $q_1, \ldots, q_k$ be the corresponding queries, and let $A_1, \ldots, A_k$ be a k-tuple of strongly consistent assignments. For each i, there is an edge of length $\frac{1}{2}$ between $\ell(r, A_1, \ldots, A_k)$ and $\ell(P_i, q_i, A_i)$.



The Graph: Vertices

- Query Vertices: For each prover P_i, for each query q_i to P_i, there is a vertex $v(P_i, q_i)$, which can only be assigned to labels corresponding to the same query of the same prover (i.e., $\ell(P_i, q_i, A)$.)

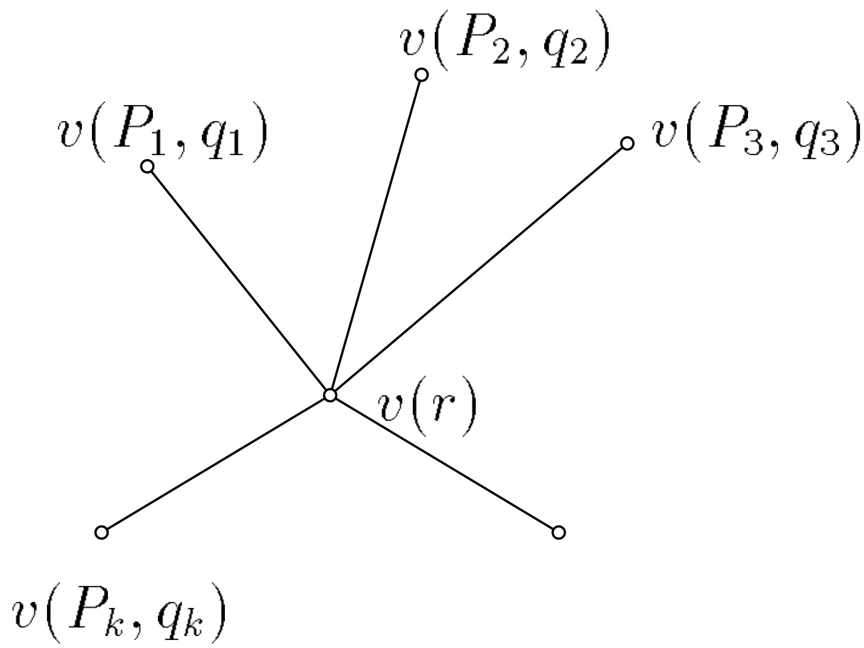
  Note that the assignments of all the query vertices to the labels define a strategy of the k provers.

- Constraint Vertices: For each random string r, there is a vertex $v(r)$, which can be only assigned to the labels corresponding to r (i.e., $\ell(r, A_1, \ldots, A_k)$).

  Note that the assignment of $v(r)$ defines the answers of the provers when the random string is r.

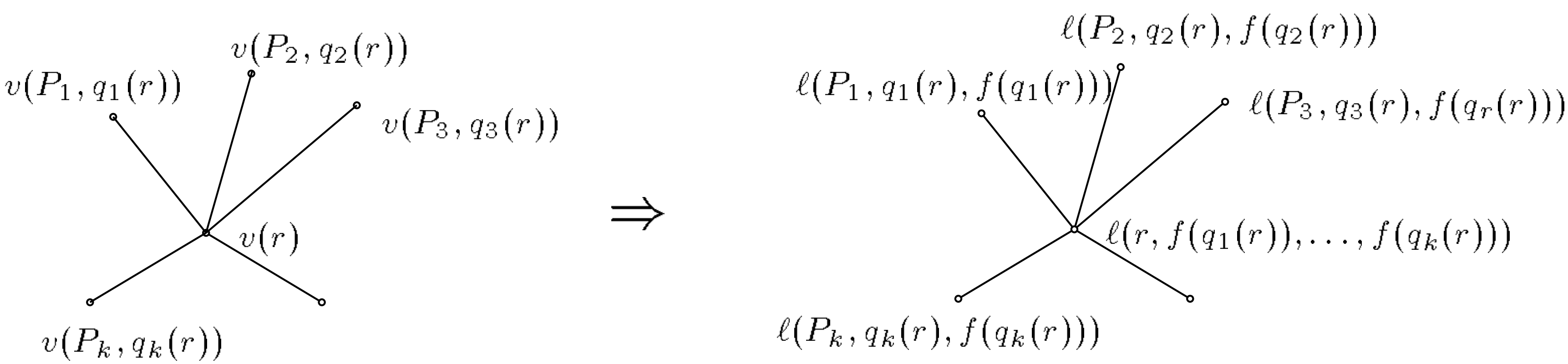
The Graph: Edges

Let $q_1, \ldots, q_k$ be the queries corresponding to random string r. Then, for each i, there is an edge between $v(r)$ and $v(P_i, q_i)$.



YES Instance

- There exists an accepting strategy of the provers.
- Queries $q_1, \ldots, q_k$ correspond to random string r.
- $A_1, \ldots, A_k$ are the answers to the queries.



Therefore, the solution cost is $\frac{k}{2}|R|$.

NO Instance

- Assignments of the query vertices define a strategy for the provers.
- Let T be the set of "inconsistent" triples (r, i, j) ($i < j$), s.t. for random string r, the answers of provers i and j are not weakly consistent.
- $|T| \geq \binom{k}{2}\frac{\varepsilon}{3}|R|$. (Recall that the probability that a pair is weakly consistent is at most $(1 - \frac{\varepsilon}{3})$).
- We can show that the solution cost is at least $|T|$, yielding a gap of $\Omega(k)$ between YES and NO instances.
- Since the construction size is $N = n^{O(k^2)}$, choosing $k = \text{poly}(\log n)$, no $\log^{\frac{1}{2}-\delta} N$ approximation exists unless NP $\subseteq$ DTIME$(n^{\text{poly} \log n})$ (for any constant δ).

Open Questions

- There is still a gap between the logarithmic upper bound and the lower bound of $\log^{1/2-\delta} n$ on the approximability of metric labeling. Can this gap be closed?

- Can we prove better (non-constant?) lower bounds on the approximability of 0-Extension?

- Or, can we obtain better approximation factors?