Metric Repair - Hardness and Approximation(s)

Qualifying Exam Presentation

Based on *Metric Violation Distance: Hardness and Approximation* [FRB22] and *Fitting Metrics and Ultrametrics With Minimum Disagreements* [CAFLDM22]

Asaf Etgar

July 2, 2024

Yale Graduate School of Arts and Science.

Talk outline

1. The Problem of Metric Repair
2. Hardness of Metric Repair
    - Decrease Only is Easy
    - Everything Else is Hard
3. Approximation Algorithms
    - First Approximation Algorithm
    - Second Approximation Algorithm
4. Future (read - current) Work

The Problem of Metric Repair

Motivation

- **Data Processing:** Many Data processing tasks (e.g, clustering) rely on the structural properties of data, such as satisfying the triangle inequality.

Motivation

- **Data Processing:** Many Data processing tasks (e.g, clustering) rely on the structural properties of data, such as satisfying the triangle inequality.
- **Graph Problems:** Hard graph problems become significantly easier when weights satisfy the triangle inequality (e.g, TSP).

Motivation

- **Data Processing:** Many Data processing tasks (e.g, clustering) rely on the structural properties of data, such as satisfying the triangle inequality.
- **Graph Problems:** Hard graph problems become significantly easier when weights satisfy the triangle inequality (e.g, TSP).
- Perhaps a good idea would be to find a metric that is ”near” the given distance measures.

Problem Definition

(K_n, w) is a complete, positively weighted graph.
M is its weighted adjacency matrix.

Problem Definition

(K_n, w) is a complete, positively weighted graph.
M is its weighted adjacency matrix.

Definition (Broken Triangle)

We say that a triangle $\{i, j, k\} \subset V$ is ***broken*** if $M_{i,j} > M_{i,k} + M_{j,k}$. In this case, we say that $\{i, j\}$ is ***heavy*** in $\{i, j, k\}$, and the other two edges are ***light***. The ***deficit*** of $\{i, j, k\}$ is $\delta(\{i, j, k\}) := M_{i,j} - (M_{i,k} + M_{j,k})$.

Definition (Metric Graph)

We say M is ***metric*** if no triangle in M is broken.

Problem Definition

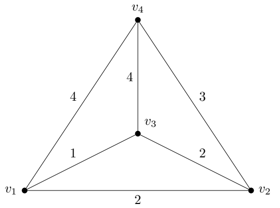
(K_n, w) is a complete, positively weighted graph.
M is its weighted adjacency matrix.

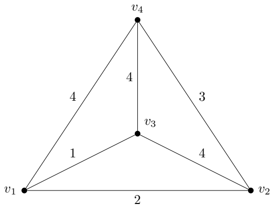
Definition (Broken Triangle)

We say that a triangle $\{i, j, k\} \subset V$ is ***broken*** if $M_{i,j} > M_{i,k} + M_{j,k}$. In this case, we say that $\{i, j\}$ is ***heavy*** in $\{i, j, k\}$, and the other two edges are ***light***. The ***deficit*** of $\{i, j, k\}$ is $\delta(\{i, j, k\}) := M_{i,j} - (M_{i,k} + M_{j,k})$.

Definition (Metric Graph)

We say M is ***metric*** if no triangle in M is broken.





Problem Definition

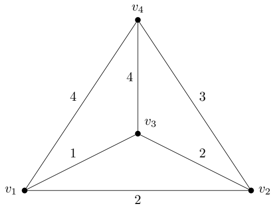
(K_n, w) is a complete, positively weighted graph.
M is its weighted adjacency matrix.

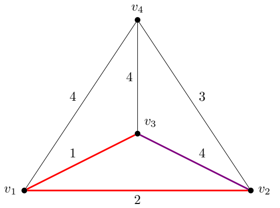
Definition (Broken Triangle)

We say that a triangle $\{i, j, k\} \subset V$ is ***broken*** if $M_{i,j} > M_{i,k} + M_{j,k}$. In this case, we say that $\{i, j\}$ is ***heavy*** in $\{i, j, k\}$, and the other two edges are ***light***. The ***deficit*** of $\{i, j, k\}$ is $\delta(\{i, j, k\}) := M_{i,j} - (M_{i,k} + M_{j,k})$.

Definition (Metric Graph)

We say M is ***metric*** if no triangle in M is broken.





Metric Repair Variants

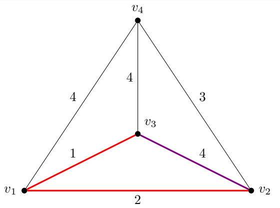
Problem (Metric Repair)

*Given (K_n, w) the problem of **Metric Repair** asks to find $w' : \binom{[n]}{2} \to \mathbb{R}_{>0}$ such that (K_n, w') is metric. A solution is **optimal** if $\|w' - w\|_0$ is minimal.*

Problem (Metric Repair)

Given (K_n, w) the problem of ***Metric Repair*** *asks to find $w' : \binom{[n]}{2} \to \mathbb{R}_{>0}$ such that (K_n, w') is metric. A solution is* ***optimal*** *if $\|w' - w\|_0$ is minimal.*

- $w' \geq w$: *Increase Only Metric Repair (**IOMR**).*
- $w' \leq w$: *Decrease Only Metric Repair (**DOMR**).*
- *Otherwise:* ***General Metric Repair*** *(**MR**).*



Hardness of Metric Repair

Decrease Only

- Intuitively speaking, DOMR is equivalent to APSP [WW10]

Decrease Only

- Intuitively speaking, DOMR is equivalent to APSP [WW10]
- Observation: If uv is heavy in a triangle, then $w(uv) \neq \mathrm{dist}_w(u, v)$.

Decrease Only

- Intuitively speaking, DOMR is equivalent to APSP [WW10]
- Observation: If uv is heavy in a triangle, then $w(uv) \neq \mathrm{dist}_w(u, v)$.
- Algorithm: [GJ17, FGR+19, FRB22]

Decrease Only

- Intuitively speaking, DOMR is equivalent to APSP [WW10]
- Observation: If uv is heavy in a triangle, then $w(uv) \neq \mathrm{dist}_w(u, v)$.
- Algorithm: [GJ17, FGR+19, FRB22]
  - Run APSP.

Decrease Only

- Intuitively speaking, DOMR is equivalent to APSP [WW10]
- Observation: If uv is heavy in a triangle, then $w(uv) \neq \mathrm{dist}_w(u,v)$.
- Algorithm: [GJ17, FGR+19, FRB22]
    - Run APSP.
    - If $w(u,v) > \mathrm{dist}_w(u,v)$, set $w'(uv) = \mathrm{dist}_w(u,v)$

Decrease Only

- Intuitively speaking, DOMR is equivalent to APSP [WW10]
- Observation: If uv is heavy in a triangle, then $w(uv) \neq \mathrm{dist}_w(u, v)$.
- Algorithm: [GJ17, FGR+19, FRB22]
  - Run APSP.
  - If $w(u, v) > \mathrm{dist}_w(u, v)$, set $w'(uv) = \mathrm{dist}_w(u, v)$
- **What about MR or IOMR?**

Theorem ([FRB22])

The decision version of MR (and IOMR) is NP complete.

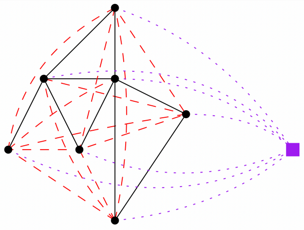
- ***k-Vertex Cover***: Given $G = (V, E)$, is there a subset $V' \subset V$ of size $|V'| \leq k$ such that every edge in G is incident with a vertex in V'?
- ***k-Metric Repair***: Is there a collection of at most k edges that changing them results in a metric graph?

We reduce $k - \mathrm{VTX} \leq_p k - \mathrm{MR}$.

Proof.

Given a graph $G = (V, E)$, we construct a complete graph $(V \cup \{v_0\}, \binom{V \cup \{v_0\}}{2})$ with weights:

$$w(u,v) = \begin{cases} 2+\varepsilon & uv \in E \\ 2 & uv \notin E \text{ and } u,v \neq v_0 \\ 1 & u = v_0 \text{ or } v = v_0 \end{cases}$$



The reduction, from [FRB22]

All broken triangles in the constructed graph must be of the form $\{v_0, u, v\}$, and every edge $uv \in E$ defines such triangle. □

Approximation Algorithms

Theorem ([FRB22])

There exists an $O\left(OPT^{1/3}\right)$ approximation algorithm for MR and IOMR that runs in $O(n^6)$ time.

Theorem ([CAFLDM22])

There exists a randomized algorithm that gives an ***expected*** *$O(\log(n))$ approximation and runs in $O(n^3)$ time. This algorithm only holds for MR.*

Definition

A collection of edges S is a ***cover*** for C if it's a hitting set for C. A cover is ***light*** if it contains a light edges from each broken cycle.

Theorem ([FGR+19])

Let C be the collection of all broken cycles in (K_n, w). Then $S \subset E$ is a valid solution to MR (IOMR) if and only if S is a (light) cover for C.

Proof.

For hard direction:

- For $xy \in S$, set $w'(xy) = \min\left\{\mathrm{dist}_{K_n \setminus S, w}(x, y), \|w\|_\infty\right\}$
- When we remove S, no broken cycles remain - so edges are shortest paths between endpoints
- so no edge that was modified to $\mathrm{dist}_{K_n \setminus S}$ is heavy.
- For edges that were set $\|w\|_\infty$ - we must have disconnected the graph when removing S. We removed at least 2 edges, and a cycle with multiple maximal weights is not broken.

□

- The theorem implies that MR is more like Set Cover - logarithmically hard to approximate.

First Approximation Algorithm

- The theorem implies that MR is more like Set Cover - logarithmically hard to approximate.
- Problem - there could be exponentially many broken cycles.

First Approximation Algorithm

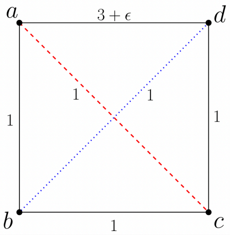
- The theorem implies that MR is more like Set Cover - logarithmically hard to approximate.
- Problem - there could be exponentially many broken cycles.
- Do we need to cover large cycles? Can't we just cover triangles?

First Approximation Algorithm

- The theorem implies that MR is more like Set Cover - logarithmically hard to approximate.
- Problem - there could be exponentially many broken cycles.
- Do we need to cover large cycles? Can’t we just cover triangles?

First Approximation Algorithm

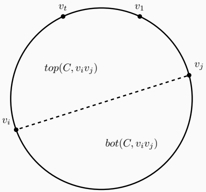
- The theorem implies that MR is more like Set Cover - logarithmically hard to approximate.
- Problem - there could be exponentially many broken cycles.
- Do we need to cover large cycles? Can't we just cover triangles?



Small cycles are not enough. [CAFLDM22]

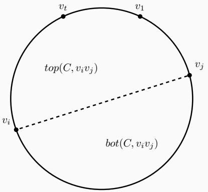
Definition

A ***unit cycle*** is a broken cycle C where for each chord e, either $bot(C, e)$ is not broken, or $bot(C, e)$ is broken with e not being the heavy edge.



$top(C, e)$ is broken for any unit cycle.

Unit Cycles



Definition

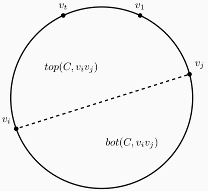
A ***unit cycle*** is a broken cycle C where for each chord e, either $bot(C, e)$ is not broken, or $bot(C, e)$ is broken with e not being the heavy edge.

$top(C, e)$ is broken for any unit cycle.

Claim

Let S be a light cover of all unit cycles in (K_n, w). Then S is a light cover of C.

Unit Cycles



Definition

A ***unit cycle*** is a broken cycle C where for each chord e, either $bot(C, e)$ is not broken, or $bot(C, e)$ is broken with e not being the heavy edge.

$top(C, e)$ is broken for any unit cycle.

Claim

Let S be a light cover of all unit cycles in (K_n, w). Then S is a light cover of C.

Proof.

If C is non-unit, then there is a chord e such that $bot(C, e)$ is broken and e is heavy. "Propagate downwards" - smallest one must be unit □

First Approximation Algorithm: Strategy

1. Cover large ***enough*** cycles, S_k.

1. Cover large ***enough*** cycles, S_k.
2. If we don't cover a large unit cycle - we cover ***important chords***.

First Approximation Algorithm: Strategy

1. Cover large ***enough*** cycles, S_k.
2. If we don't cover a large unit cycle - we cover ***important chords***.
3. Use cycles ***induced*** by chords to cover larger unit cycles S_c.

First Approximation Algorithm: Strategy

1. Cover large ***enough*** cycles, S_k.
2. If we don't cover a large unit cycle - we cover ***important chords***.
3. Use cycles ***induced*** by chords to cover larger unit cycles S_c.
4. return $S_c \cup S_k$.

Claim

Let $\mathcal{C}_{\leq k}$ be the collection of all broken cycles of size at most k in (K_n, w). Then one can compute a light cover S_k for $\mathcal{C}_{\leq k}$ in $O(n^k)$ time, and $|S_k| \leq (k-1)|opt_k|$.

First Approximation Algorithm: Cover small cycles

Claim

Let $\mathcal{C}_{\leq k}$ be the collection of all broken cycles of size at most k in (K_n, w). Then one can compute a light cover S_k for $\mathcal{C}_{\leq k}$ in $O(n^k)$ time, and $|S_k| \leq (k-1)|opt_k|$.

Proof.

We use a standard Hitting Set approximation for bounded size sets:

- Every $C \in \mathcal{C}_{\leq k}$ defines a collection of $k-1$ light edges $\ell(C)$.

Claim

Let $\mathcal{C}_{\leq k}$ be the collection of all broken cycles of size at most k in (K_n, w). Then one can compute a light cover S_k for $\mathcal{C}_{\leq k}$ in $O(n^k)$ time, and $|S_k| \leq (k-1)|opt_k|$.

Proof.

We use a standard Hitting Set approximation for bounded size sets:

- Every $C \in \mathcal{C}_{\leq k}$ defines a collection of $k-1$ light edges $\ell(C)$.
- Set $S \leftarrow \emptyset$.

Claim

Let $\mathcal{C}_{\leq k}$ be the collection of all broken cycles of size at most k in (K_n, w). Then one can compute a light cover S_k for $\mathcal{C}_{\leq k}$ in $O(n^k)$ time, and $|S_k| \leq (k-1)|opt_k|$.

Proof.

We use a standard Hitting Set approximation for bounded size sets:

- Every $C \in \mathcal{C}_{\leq k}$ defines a collection of $k-1$ light edges $\ell(C)$.
- Set $S \leftarrow \emptyset$.
- Go over $(\ell(C))_{C \in \mathcal{C}}$ one by one. If $S \cap \ell(C) = \emptyset, S \leftarrow S \cup \ell(C)$.

Claim

Let $\mathcal{C}_{\leq k}$ be the collection of all broken cycles of size at most k in (K_n, w). Then one can compute a light cover S_k for $\mathcal{C}_{\leq k}$ in $O(n^k)$ time, and $|S_k| \leq (k-1)|opt_k|$.

Proof.

We use a standard Hitting Set approximation for bounded size sets:

- Every $C \in \mathcal{C}_{\leq k}$ defines a collection of $k-1$ light edges $\ell(C)$.
- Set $S \leftarrow \emptyset$.
- Go over $(\ell(C))_{C \in \mathcal{C}}$ one by one. If $S \cap \ell(C) = \emptyset, S \leftarrow S \cup \ell(C)$.

First Approximation Algorithm: Cover small cycles

Claim

Let $\mathcal{C}_{\leq k}$ be the collection of all broken cycles of size at most k in (K_n, w). Then one can compute a light cover S_k for $\mathcal{C}_{\leq k}$ in $O(n^k)$ time, and $|S_k| \leq (k-1)|opt_k|$.

Proof.

We use a standard Hitting Set approximation for bounded size sets:

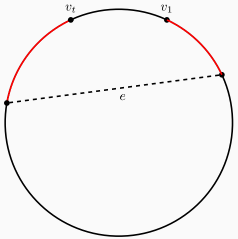
- Every $C \in \mathcal{C}_{\leq k}$ defines a collection of $k-1$ light edges $\ell(C)$.
- Set $S \leftarrow \emptyset$.
- Go over $(\ell(C))_{C \in \mathcal{C}}$ one by one. If $S \cap \ell(C) = \emptyset, S \leftarrow S \cup \ell(C)$.

$|S_k| \leq (k-1)|opt_k|$ since whenever we encounter an uncovered set, any solution would add at least one edge to opt_6, while we add at most $(k-1)$. □

Corollary

Let S_k be as before, and C be a unit cycle uncovered by S_k. Let e be a chord of C such that $|top(C, e)| \leq k$. Then $e \in S_k$.

Proof.

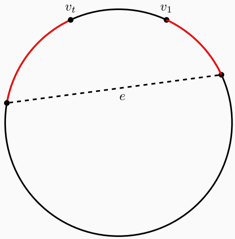


□

Corollary

Let S_k be as before, and C be a unit cycle uncovered by S_k. Let e be a chord of C such that $|top(C, e)| \leq k$. Then $e \in S_k$.

Proof.

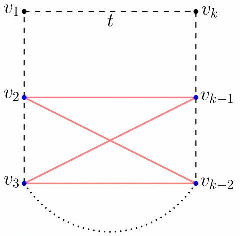


- *We need to cover uncovered unit cycles of size at least k.*
- *Each such cycle has many cords in S_k.*
- *These chords induce cycles that help us!*

$\square$

From now on $k = 6$

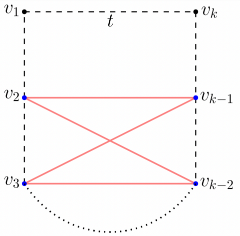
For a broken cycle C with $|C| > 6$ and heavy edge $v_1 v_k$, and consider the edge-induced 4-cycle closest to (but not touching) $v_1 v_k$



$embed4(C)$ [FRB22]

From now on $k = 6$

For a broken cycle C with $|C| > 6$ and heavy edge $v_1 v_k$, and consider the edge-induced 4-cycle closest to (but not touching) $v_1 v_k$



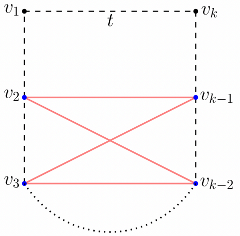
$embed4(C)$ [FRB22]

Observation: chords of this cycle are light edges of C. Moreover, these cycles appear in S_6 for each uncovered unit cycle C.

Lemma

Let S_6 be as before, and let S_c be a set of edges containing a chord of every 4-cycle induced by S_6. Then $S_6 \cup S_c$ is a solution to IOMR.

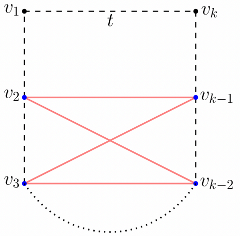
First Approximation Algorithm: Chording 4-cycles



$embed4(C)$ [FRB22]

We find a chord- cover for all 4-cycles in S_6:

1. $S_c \leftarrow \emptyset$
2. For each 4-cycle $v_2 v_{k-1} v_3 v_{k-2}$ in S_6, if $v_2 v_3 \notin S_c$ and $v_{k-1} v_{k-2} \notin S_c{}^a$, $S_c \leftarrow S_c \cup \{v_{k-1}v_{k-2}, v_2 v_3\}$



$embed4(C)$ [FRB22]

We find a chord- cover for all 4-cycles in S_6:

1. $S_c \leftarrow \emptyset$
2. For each 4-cycle $v_2 v_{k-1} v_3 v_{k-2}$ in S_6, if $v_2 v_3 \notin S_c$ and $v_{k-1} v_{k-2} \notin S_c$ [a], $S_c \leftarrow S_c \cup \{v_{k-1} v_{k-2}, v_2 v_3\}$

We know $|S_6| \leq 5|opt_5| \leq 5|OPT|$. If we can bound $|S_c|$ as a function of $|S_6|$, we will get an approximation.

[a] $v_{k-1} v_{k-2}, v_2 v_3$ need not be in S_6

Let $G = (V, S_6)$ be the graph induced by the edges in S_6, and let $\tilde{C}$ be the collection of cycle from which we added edges to S_c.

- No two cycles in $\tilde{C}$ can share a chord, so $|S_c|/2 = |\tilde{C}|$

Let $G = (V, S_6)$ be the graph induced by the edges in S_6, and let $\tilde{C}$ be the collection of cycle from which we added edges to S_c.

- No two cycles in $\tilde{C}$ can share a chord, so $|S_c|/2 = |\tilde{C}|$
- Clearly $|E(\tilde{C})| \leq |S_6|$.

Let $G = (V, S_6)$ be the graph induced by the edges in S_6, and let $\tilde{C}$ be the collection of cycle from which we added edges to S_c.

- No two cycles in $\tilde{C}$ can share a chord, so $|S_c|/2 = |\tilde{C}|$
- Clearly $|E(\tilde{C})| \leq |S_6|$.
- It sufficed to bound $|\tilde{C}|$ in terms of $|E(\tilde{C})|$.

First Approximation Algorithm: Chording 4-cycles: Not too many edges

Let $G = (V, S_6)$ be the graph induced by the edges in S_6, and let $\tilde{C}$ be the collection of cycle from which we added edges to S_c.

- No two cycles in $\tilde{C}$ can share a chord, so $|S_c|/2 = |\tilde{C}|$
- Clearly $|E(\tilde{C})| \leq |S_6|$.
- It sufficed to bound $|\tilde{C}|$ in terms of $|E(\tilde{C})|$.

First Approximation Algorithm: Chording 4-cycles: Not too many edges

Let $G = (V, S_6)$ be the graph induced by the edges in S_6, and let $\tilde{C}$ be the collection of cycle from which we added edges to S_c.

- No two cycles in $\tilde{C}$ can share a chord, so $|S_c|/2 = |\tilde{C}|$
- Clearly $|E(\tilde{C})| \leq |S_6|$.
- It sufficed to bound $|\tilde{C}|$ in terms of $|E(\tilde{C})|$.

Lemma

Let $G = (V, E)$ be a graph whose edge set comprised of a collection of 4-cycles $\tilde{C}$, no two of which share a chord. Then $|\tilde{C}| = O(|E|^{\frac{4}{3}})$.

Proof

Algorithm 1 Approximate IOMR

Input: K_n, w

1: Compute S_6 using standard HS approximation.
2: Compute S_c using S_6 as described before.
3: **return** $S_6 \cup S_c$.

Theorem

Algorithm 1 is an $O(OPT^{\frac{1}{3}})$ approximation to IOMR.

Proof.

OPT is the size of a minimum HS for C. Clearly $|opt_5| \leq OPT$, and so $|S_6| = O(OPT)$. By the previous lemma, $|S_c| = O(|S_6|^{\frac{4}{3}}) = O(OPT^{\frac{4}{3}})$. □

Second Approximation Algorithm: Intro

- *Ransomized* algorithm.
- $\log(n)$ approximation ***in expectation*** (exponentially better!)
- $O(n^3)$ runtime.
- Cannot be (naively) modified to solve IOMR.

Second Approximation Algorithm: Intuition

- If (K_n, w) is metric, then there are no broken triangle.
- Conversely - if there is a broken triangle, (K_n, w) is not metric.

Algorithm 2 Pivot

Input: K_n, w, i

```
for j, k ∈ ([n]\i choose 2) do
    if w(jk) > w(ij) + w(ik) then
        w(jk) = w(ij) + w(ik)
    if w(jk) < |w(ij) − w(ik)| then
        w(jk) = |w(ij) − w(ik)|
```

Claim

After pivoting at i, no broken triangles incident to i remain.

Algorithm 3 Randomized IOMR

Input: K_n, w

1: Pick $i \in [n]$ uniformly at random
2: Pivot at i
3: Call Randomized IOMR on $K_n \setminus \{i\}, w\mid_{[n]\setminus\{i\}}$

After pivoting at i, we don't change any edge incident to i anymore.

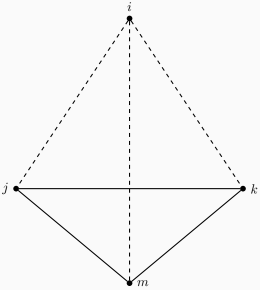
Lemma

After pivoting at i, no new broken triangles are created.

Lemma

After pivoting at i, no new broken triangles are created.

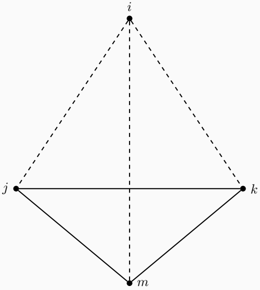
Proof.

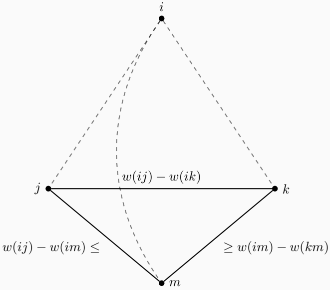


Lemma

After pivoting at i, no new broken triangles are created.

Proof.

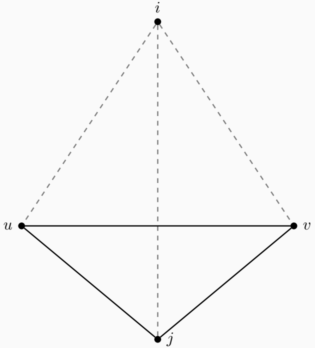




$\square$

Lemma

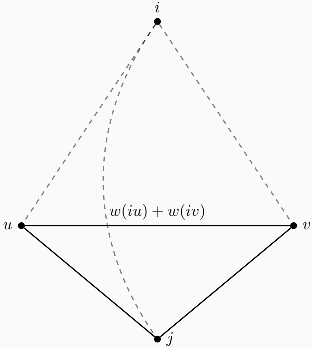
Let uv be an edge that is adjacent to k broken triangles, in which uv is heavy. Order the third vertex of said triangles in order $1, \ldots k$ such that $\delta(uvi) \geq \delta(uvj)$ for $j \geq i$. Then pivoting at i fixes $\{u, v, j\}$ for all $j \geq i$.



Second Approximation Algorithm: Approximation intuition

Lemma

Let uv be an edge that is adjacent to k broken triangles, in which uv is heavy. Order the third vertex of said triangles in order $1, \dots k$ such that $\delta(uvi) \geq \delta(uvj)$ for $j \geq i$. Then pivoting at i fixes $\{u, v, j\}$ for all $j \geq i$.



w' are the weights after pivoting at i.

1. $w'(uv) \leq w'(ju) + w'(jv)$:
   - $\delta_i \leq \delta_j$, so $w(iu) + w(iv) \leq w(ju) + w(jv)$.
   - WLOG $w'(uj) = w(ui) + w(ij)$
   - But $w'(vj) \geq w(vi) - w(ij)$
2. $w'(uv) \geq |w'(ju) - w'(jv)|$.
   - $w(iu) + w(iv) < w'(ju) - w'(jv)$.
   - Since $w'(uj) \leq w(ij) + w(iu)$ and $w(ij) \leq w(iv) + w'(jv)$, we obtain contradiction.

- The algorithm makes progress
- The algorithm fixes ”about half” of the broken triangles incident to any edge at each step

How do we formalize this? Let $\mathcal{T}$ be the collection of triangles, and $\mathcal{T}'$ the broken triangles.

- Every solution to MR contains a ***Hitting Set*** for $\mathcal{T}'$

Second Approximation Algorithm: Summary

- The algorithm makes progress
- The algorithm fixes "about half" of the broken triangles incident to any edge at each step

How do we formalize this? Let $\mathcal{T}$ be the collection of triangles, and $\mathcal{T}'$ the broken triangles.

- Every solution to MR contains a ***Hitting Set*** for $\mathcal{T}'$
- So $|OPT_{MR}| \geq |HS_{\mathcal{T}'}|$

Second Approximation Algorithm: Summary

- The algorithm makes progress
- The algorithm fixes ”about half” of the broken triangles incident to any edge at each step

How do we formalize this? Let $\mathcal{T}$ be the collection of triangles, and $\mathcal{T}'$ the broken triangles.

- Every solution to MR contains a ***Hitting Set*** for $\mathcal{T}'$
- So $|OPT_{MR}| \geq |HS_{\mathcal{T}'}|$
- By duality - $|HS_{\mathcal{T}'}| \geq |HS^f_{\mathcal{T}'}| \geq |PACK^f_{\mathcal{T}'}|$

Second Approximation Algorithm: Summary

- The algorithm makes progress
- The algorithm fixes ”about half” of the broken triangles incident to any edge at each step

How do we formalize this? Let $\mathcal{T}$ be the collection of triangles, and $\mathcal{T}'$ the broken triangles.

- Every solution to MR contains a ***Hitting Set*** for $\mathcal{T}'$
- So $|OPT_{MR}| \geq |HS_{\mathcal{T}'}|$
- By duality - $|HS_{\mathcal{T}'}| \geq |HS^f_{\mathcal{T}'}| \geq |PACK^f_{\mathcal{T}'}|$

Second Approximation Algorithm: Summary

- The algorithm makes progress
- The algorithm fixes ”about half” of the broken triangles incident to any edge at each step

How do we formalize this? Let $\mathcal{T}$ be the collection of triangles, and $\mathcal{T}'$ the broken triangles.

- Every solution to MR contains a ***Hitting Set*** for $\mathcal{T}'$
- So $|OPT_{MR}| \geq |HS_{\mathcal{T}'}|$
- By duality - $|HS_{\mathcal{T}'}| \geq |HS^f_{\mathcal{T}'}| \geq |PACK^f_{\mathcal{T}'}|$

Where the Fractional Packing problem is

$$\max_{p \in [0,1]^{|\mathcal{T}'|}} \sum_{t \in \mathcal{T}'} p_t \quad \text{s.t} \quad \forall e \in \binom{[n]}{2}, \sum_{t \ni e} p_t \leq 1$$

If we find some α and p_t such that $\sum_{t \ni e} \frac{p_t}{\alpha} \leq 1$, we have an α approximation

Second Approximation Algorithm: Proof

- $ALG :=$ the set of edges augmented by the algorithm.

Second Approximation Algorithm: Proof

- $ALG :=$ the set of edges augmented by the algorithm.
- $A_{i,j,k} :=$ the event that one of $\{i, j, k\}$ was chosen as pivot, and $\{i, j, k\}$ was modified as a result. $p_t = \mathbb{E}[A_t]$.

Second Approximation Algorithm: Proof

- $ALG :=$ the set of edges augmented by the algorithm.
- $A_{i,j,k} :=$ the event that one of $\{i, j, k\}$ was chosen as pivot, and $\{i, j, k\}$ was modified as a result. $p_t = \mathbb{E}[A_t]$.
- $\mathbb{E}[|ALG|] \leq \sum_{t \in \mathcal{T}} p_t = \sum_{t \in \mathcal{T}'} p_t.$

Fix e.

- When A_t happens, it changes e with probability 1/3, so $\alpha(e) := \mathbb{E}\,[\#\text{times } e \text{ was modified}] = \sum_{t \ni e} p_t/3$.
- We induct on n', the number of broken triangles e is incident to.
- Let $c_e(n')$ be an upper bound on the expected number of modifications of e when it is incident to n' broken triangles.
- Any partitioning $n_1 + n_2 = n'$ corresponds to possible increase/decrease of e when pivoting.
- We use conditional probability on all such partitions, and later condition on "how big" the deficit was when we pivot.
- "Isoperimetric" qualities simplify our computation, as well as Stirling's approximation.

Details

A note on Ultrametrics

- An ***Ultrametric*** is a metric d with the property that

$$d(i,j) \leq \max d(i,k), d(j,k)$$

A note on Ultrametrics

- An ***Ultrametric*** is a metric d with the property that

$$d(i,j) \leq \max d(i,k), d(j,k)$$

- Because of this additional structure - possible to get a better approximation ratio. Namely, $O(1)$ approximation!

A note on Ultrametrics

- An ***Ultrametric*** is a metric d with the property that

$$d(i,j) \leq \max d(i,k), d(j,k)$$

- Because of this additional structure - possible to get a better approximation ratio. Namely, $O(1)$ approximation!
- The idea is to use ***correlation clustering*** in a sophisticated way (because of ball structure of the metric).

Future (read - current) Work

Open Problems

- ***Generalized*** metric repair - $(G = (V, E), w)$ where triangle inequalities are ***cycle*** inequalities.

Open Problems

- ***Generalized*** metric repair - $(G = (V, E), w)$ where triangle inequalities are ***cycle*** inequalities.
- Derandomization?

Open Problems

- ***Generalized*** metric repair - $(G = (V, E), w)$ where triangle inequalities are ***cycle*** inequalities.
- Derandomization?
- ”Average case” graphs

Open Problems

- ***Generalized*** metric repair - $(G = (V, E), w)$ where triangle inequalities are ***cycle*** inequalities.
- Derandomization?
- ”Average case” graphs

- ***Generalized*** metric repair - $(G = (V, E), w)$ where triangle inequalities are ***cycle*** inequalities.
- Derandomization?
- ”Average case” graphs

Thank You! Questions?

Second Approximation Algorithm: Proof

For n_1, n_2 such that $n_1 + n_2 = n'$, let E_{n_1,n_2} be the event that pivoting at n_1 of the triangles induces an increase, and n_2 induces a decrease. Conditioned on E_{n_1,n_2}, for $k \in [n_1]$ let F_{k,n_2} be the event that the pivot chosen induced the kth largest increase (similarly define $G_{n_1,k}$). Back

$$c_e(n') \leq$$

$$\leq 1 + \sum_{i \in [n']} c_e(i) \Pr\left[i \text{ broken triangles remain after pivoting}\right] \leq$$

Second Approximation Algorithm: Proof

$$c_e(n') \leq$$

$$\leq 1 + \sum_{i \in [n']} c_e(i) \Pr\left[i \text{ broken triangles remain after pivoting}\right] \leq$$

$$\leq 1 + \sum_{n_1, n_2} \Pr\left[E_{n_1,n_2}\right] \left(\sum_{k=1}^{n_1} \Pr\left[F_{k,n_2}\right] c_e(n_2 + k - 1) + \sum_{k'=1}^{n_2} \Pr\left[G_{n_2,k'}\right] c_e(n_1 + k' - 1) \right)$$

Back

$$c_e(n') \leq$$

$$\leq 1 + \sum_{i \in [n']} c_e(i) \Pr\left[i \text{ broken triangles remain after pivoting}\right] \leq$$

$$\leq 1 + \sum_{n_1, n_2} \Pr\left[E_{n_1,n_2}\right] \left(\sum_{k=1}^{n_1} \Pr\left[F_{k,n_2}\right] c_e(n_2 + k - 1) + \sum_{k'=1}^{n_2} \Pr\left[G_{n_2,k'}\right] c_e(n_1 + k' - 1)\right)$$

$$\leq 1 + \max_{n_1 + n_2 = n'} \left(\sum_{k=1}^{n_1} \frac{1}{n'} c_e(n_1 + k - 1) + \sum_{k'=1}^{n_2} \frac{1}{n'} c_e(n_2 + k' - 1)\right)$$

Second Approximation Algorithm: Proof

$$
\begin{aligned}
& c_e(n') \leq \\
& \leq 1 + \sum_{i \in [n']} c_e(i) \Pr\left[i \text{ broken triangles remain after pivoting}\right] \leq \\
& \leq 1 + \sum_{n_1, n_2} \Pr\left[E_{n_1,n_2}\right] \left(\sum_{k=1}^{n_1} \Pr\left[F_{k,n_2}\right] c_e(n_2 + k - 1) + \sum_{k'=1}^{n_2} \Pr\left[G_{n_2,k'}\right] c_e(n_1 + k' - 1)\right) \\
& \leq 1 + \max_{n_1+n_2=n'} \left(\sum_{k=1}^{n_1} \frac{1}{n'} c_e(n_1 + k - 1) + \sum_{k'=1}^{n_2} \frac{1}{n'} c_e(n_2 + k' - 1)\right) \\
& \leq 1 + \max_{n_1+n_2=n'} \left(\sum_{k=n_1+1}^{n'} \frac{1}{n'} c_e(k - 1) + \sum_{k'=n_2+1}^{n'} \frac{1}{n'} c_e(k' - 1)\right)
\end{aligned}
$$

Back

Second Approximation Algorithm: Proof

$$
\begin{aligned}
&c_e(n') \leq \\
&\leq 1 + \sum_{i \in [n']} c_e(i) \Pr\left[i \text{ broken triangles remain after pivoting}\right] \leq \\
&\leq 1 + \sum_{n_1, n_2} \Pr\left[E_{n_1,n_2}\right] \left(\sum_{k=1}^{n_1} \Pr\left[F_{k,n_2}\right] c_e(n_2 + k - 1) + \sum_{k'=1}^{n_2} \Pr\left[G_{n_2,k'}\right] c_e(n_1 + k' - 1)\right) \\
&\leq 1 + \max_{n_1+n_2=n'} \left(\sum_{k=1}^{n_1} \frac{1}{n'} c_e(n_1 + k - 1) + \sum_{k'=1}^{n_2} \frac{1}{n'} c_e(n_2 + k' - 1)\right) \\
&\leq 1 + \max_{n_1+n_2=n'} \left(\sum_{k=n_1+1}^{n'} \frac{1}{n'} c_e(k - 1) + \sum_{k'=n_2+1}^{n'} \frac{1}{n'} c_e(k' - 1)\right) \\
&\leq 1 + \sum_{k=\lfloor n' \rfloor+1}^{n'} \frac{1}{n'} c_e(k - 1) + \sum_{k=\lceil n' \rceil+1}^{n'} \frac{1}{n'} c_e(k - 1)
\end{aligned}
$$

Back

$$
\begin{aligned}
&c_e(n') \leq \\
&\leq 1 + \sum_{i \in [n']} c_e(i) \Pr\left[i \text{ broken triangles remain after pivoting}\right] \leq \\
&\leq 1 + \sum_{n_1, n_2} \Pr\left[E_{n_1,n_2}\right] \left(\sum_{k=1}^{n_1} \Pr\left[F_{k,n_2}\right] c_e(n_2 + k - 1) + \sum_{k'=1}^{n_2} \Pr\left[G_{n_2,k'}\right] c_e(n_1 + k' - 1) \right) \\
&\leq 1 + \max_{n_1+n_2=n'} \left(\sum_{k=1}^{n_1} \frac{1}{n'} c_e(n_1 + k - 1) + \sum_{k'=1}^{n_2} \frac{1}{n'} c_e(n_2 + k' - 1) \right) \\
&\leq 1 + \max_{n_1+n_2=n'} \left(\sum_{k=n_1+1}^{n'} \frac{1}{n'} c_e(k - 1) + \sum_{k'=n_2+1}^{n'} \frac{1}{n'} c_e(k' - 1) \right) \\
&\leq 1 + \sum_{k=\lfloor n' \rfloor + 1}^{n'} \frac{1}{n'} c_e(k - 1) + \sum_{k=\lceil n' \rceil + 1}^{n'} \frac{1}{n'} c_e(k - 1) \\
&\leq 1 + \frac{1}{n'} c \ln \left(\frac{(n'!)^2}{\lfloor n' \rfloor ! \lceil n' \rceil !} \right) \leq c \ln(n' + 1)
\end{aligned}
$$

Back

Lemma

Let $G = (V, E)$ be a graph whose edge set comprised of a collection of 4-cycles $\tilde{C}$, no two of which share a chord. Then $|\tilde{C}| = O(|E|^{\frac{4}{3}})$.

Proof.

- Let V_s be the vertices in G of degree $\leq |E|^{\frac{1}{3}}$ and $V_l := V \setminus V_s$.
- Partition V_s into sets $(g_i)_{i=1}^{k}$ such that $|E|^{\frac{1}{3}} \leq \sum_{v \in g_i} \deg(v) \leq 2|E|^{\frac{1}{3}}$.
- By degree-sum, $k|E|^{\frac{1}{3}} \leq 2|E|$ so $k \leq 2|E|^{\frac{2}{3}}$.
- Any two edges incident to v can appear in at most one 4 cycle, hence the number of 4 cycles v is in is bounded by $\binom{\deg(v)}{2} \leq \frac{\deg(v)^2}{2}$
- So total number of cycles is bounded by $\frac{1}{2}\sum_{i=1}^{k}\sum_{v \in g_i} \deg(v)^2 \leq \frac{1}{2}\sum_{i=1}^{k}\left(\sum_{v \in g_i} \deg(v)\right)^2 \leq k4|E|^{\frac{2}{3}} \leq 4|E|^{\frac{4}{3}}$
- By similar reasoning, $|V_l||E|^{\frac{1}{3}} \leq 2|E|$ and V_l can define at most $|E|^{\frac{4}{3}}$ cycles.

$\square$

Back

References

- Vincent Cohen-Addad, Chenglin Fan, Euiwoong Lee, and Arnaud De Mesmay, ***Fitting metrics and ultrametrics with minimum disagreements***, 2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS), IEEE, 2022, pp. 301–311.
- Chenglin Fan, Anna C Gilbert, Benjamin Raichel, Rishi Sonthalia, and Gregory Van Buskirk, ***Generalized metric repair on graphs***, arXiv preprint arXiv:1908.08411 (2019).
- Chenglin Fan, Benjamin Raichel, and Gregory Van Buskirk, ***Metric violation distance: Hardness and approximation***, Algorithmica **84** (2022), no. 5, 1441–1465.
- Anna C Gilbert and Lalit Jain, ***If it ain't broke, don't fix it: Sparse metric repair***, 2017 55th Annual Allerton Conference on Communication, Control, and Computing (Allerton), IEEE, 2017, pp. 612–619.
- Virginia Vassilevska Williams and Ryan Williams, ***Subcubic equivalences between path, matrix and triangle problems***, 2010 IEEE 51st Annual Symposium on Foundations of Computer Science, 2010, pp. 645–654.