On Minimum Sum of Radii and Diameters Clustering

Babak Behsaz Mohammad R. Salavatipour

Department of Computing Science
University of Alberta

July 4, 2012

Problem Statement

Minimum Sum of Radii (MSR) and Diameters (MSD) Problem

- **Input:** a metric (V, d): can be seen as an edge-weighted complete graph, an integer k.

Problem Statement

Minimum Sum of Radii (MSR) and Diameters (MSD) Problem

- **Input:** a metric (V, d): can be seen as an edge-weighted complete graph, an integer k.
- **Goal:** partition the points of V into at most k clusters $V_1, V_2, \ldots, V_k$.

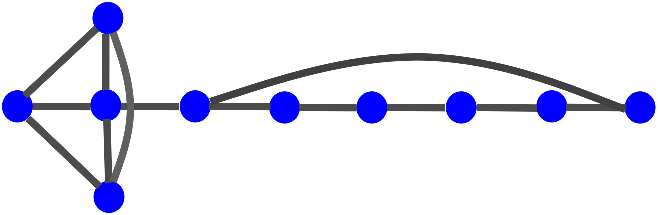
Problem Statement

Minimum Sum of Radii (MSR) and Diameters (MSD) Problem

- **Input:** a metric (V, d): can be seen as an edge-weighted complete graph, an integer k.
- **Goal:** partition the points of V into at most k clusters $V_1, V_2, \ldots, V_k$.
- **Objective:** minimize $\sum_{i=1}^{k} \text{rad}(V_i)$ in MSR, minimize $\sum_{i=1}^{k} \text{diam}(V_i)$ in MSD.
- **Radius and Diameter:** $\text{rad}(V_i) = \min_{u \in V_i} \max_{v \in V_i} d(u, v)$, $\text{diam}(V_i) = \max_{u,v \in V_i} d(u, v)$

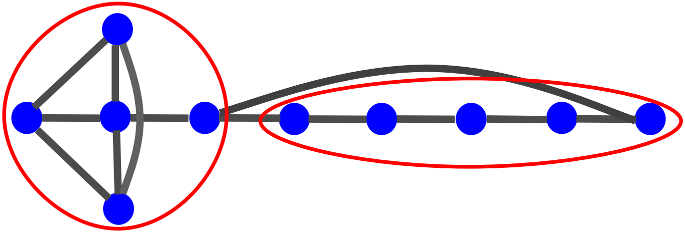
An Example of MSR and MSD

Input: G=the metric completion of the following graph, $k = 2$.



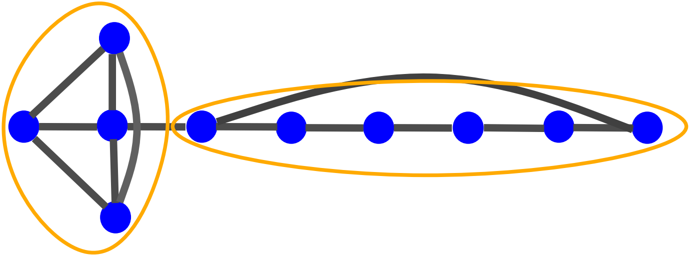
An Example of MSR and MSD

Input: G=the metric completion of the following graph, $k = 2$.
Solution 1: MSR objective $= 1 + 2$ and MSD objective $= 2 + 3$.



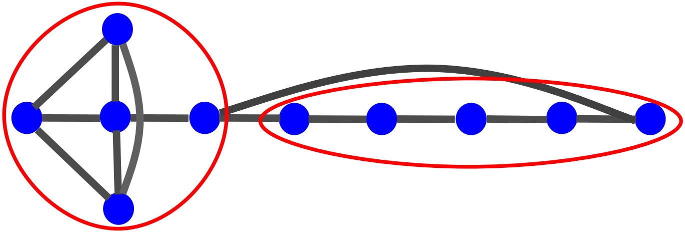
An Example of MSR and MSD

Input: G=the metric completion of the following graph, $k = 2$.
Solution 1: MSR objective $= 1 + 2$ and MSD objective $= 2 + 3$.
Solution 2: MSR objective $= 1 + 3$ and MSD objective $= 1 + 3$.



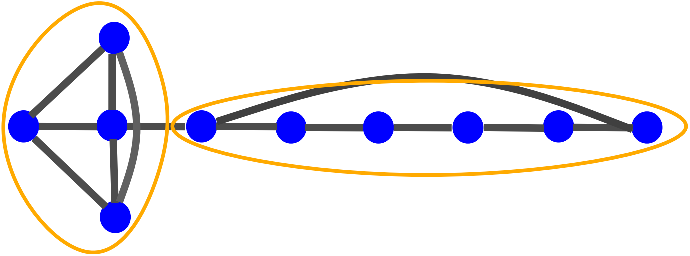
An Example of MSR and MSD

Input: G=the metric completion of the following graph, $k = 2$.
Solution 1: **MSR objective =** $1 + 2$ and MSD objective $= 2 + 3$.
Solution 2: MSR objective $= 1 + 3$ and MSD objective $= 1 + 3$.



An Example of MSR and MSD

Input: G=the metric completion of the following graph, $k = 2$.
Solution 1: MSR objective $= 1 + 2$ and MSD objective $= 2 + 3$.
Solution 2: MSR objective $= 1 + 3$ and **MSD objective =** $1 + 3$.



Motivations

Clustering

Improving the k-center clustering.

Motivations

Clustering

Improving the k-center clustering.

Communication Networks

Location of base stations in a wireless data network.

MSR and MSD Previous Works: the general case

Hardness results:

- MSD is $(2-\epsilon)$-hard.

MSR and MSD Previous Works: the general case

Hardness results:

- MSD is $(2 - \epsilon)$-hard.
- MSR is NP-hard.

MSR and MSD Previous Works: the general case

Hardness results:

- MSD is $(2-\epsilon)$-hard.
- MSR is NP-hard.

Algorithmic results:

- Observation: an α-approximation for MSR (MSD) $\rightarrow$ a (2α)-approximation for MSD (MSR)

MSR and MSD Previous Works: the general case

Hardness results:

- MSD is $(2-\epsilon)$-hard.
- MSR is NP-hard.

Algorithmic results:

- Observation: an α-approximation for MSR (MSD) $\rightarrow$ a (2α)-approximation for MSD (MSR)
- 3.504-approximation for MSR (Charikar-Panigrahy, STOC'01)

MSR and MSD Previous Works: the general case

Hardness results:

- MSD is $(2-\epsilon)$-hard.
- MSR is NP-hard.

Algorithmic results:

- Observation: an α-approximation for MSR (MSD) $\rightarrow$ a (2α)-approximation for MSD (MSR)
- 3.504-approximation for MSR (Charikar-Panigrahy, STOC'01) $\rightarrow$ 7.008-approximation for MSD

MSR and MSD Previous Works: the general case

Hardness results:

- MSD is $(2-\epsilon)$-hard.
- MSR is NP-hard.

Algorithmic results:

- Observation: an α-approximation for MSR (MSD) $\rightarrow$ a (2α)-approximation for MSD (MSR)
- 3.504-approximation for MSR (Charikar-Panigrahy, STOC'01) $\rightarrow$ 7.008-approximation for MSD
- exact algorithm for MSR in time $n^{O(\log n \log \Delta)}$ where Δ is the ratio of largest distance over the smallest distance (Gibson *et al.*, SWAT'08) $\rightarrow$ QPTAS for MSR

Previous Works: the special cases

- MSD, $k = 2$: exact algorithm (Hansen-Jaumard, J. of Classification'87)

Previous Works: the special cases

- MSD, $k = 2$: exact algorithm (Hansen-Jaumard, J. of Classification'87)
- Euclidean MSD, constant k: exact algorithm. (Capoyleas-Rote-Woeginger, J. of Algorithms'91)

Previous Works: the special cases

- MSD, $k = 2$: exact algorithm (Hansen-Jaumard, J. of Classification'87)
- Euclidean MSD, constant k: exact algorithm. (Capoyleas-Rote-Woeginger, J. of Algorithms'91)
- General MSD, constant k: 2-approximation $\leftarrow$ comes from an exact algorithm for MSR. (Doddi *et al.*, SWAT'00 and Nordic J. of Computing'00)

Previous Works: the special cases

- MSD, $k = 2$: exact algorithm (Hansen-Jaumard, J. of Classification'87)
- Euclidean MSD, constant k: exact algorithm. (Capoyleas-Rote-Woeginger, J. of Algorithms'91)
- General MSD, constant k: 2-approximation $\leftarrow$ comes from an exact algorithm for MSR. (Doddi *et al.*, SWAT'00 and Nordic J. of Computing'00)
- Euclidean MSR: exact algorithm $\rightarrow$ a 2-approximation for Euclidean MSD. (Gibson *et al.*, SODA'08)

Overview of main results

Metrics with polynomially bounded Δ: exact algorithm for MSR in time $n^{O(\log^2 n)}$ $\rightarrow$ exact algorithm for MSR in this case?

Overview of main results

Metrics with polynomially bounded Δ: exact algorithm for MSR in time $n^{O(\log^2 n)}$ $\rightarrow$ exact algorithm for MSR in this case?

Theorem

There is a polynomial time exact algorithm for the unweighted MSR problem when no clusters of radius zero) is allowed.

Overview of main results

Metrics with polynomially bounded Δ: exact algorithm for MSR in time $n^{O(\log^2 n)}$ $\rightarrow$ exact algorithm for MSR in this case?

Theorem

There is a polynomial time exact algorithm for the unweighted MSR problem when no clusters of radius zero) is allowed.

- Euclidean MSD: exact algorithm for constant k. Euclidean MSD with variable k?

Overview of main results

Metrics with polynomially bounded Δ: exact algorithm for MSR in time $n^{O(\log^2 n)}$ $\rightarrow$ exact algorithm for MSR in this case?

Theorem

There is a polynomial time exact algorithm for the unweighted MSR problem when no clusters of radius zero) is allowed.

- Euclidean MSD: exact algorithm for constant k. Euclidean MSD with variable k?
- 2-approximation for Euclidean MSD + ratio 2 hardness for general MSD. Can we beat factor 2?

Overview of main results

Metrics with polynomially bounded Δ: exact algorithm for MSR in time $n^{O(\log^2 n)}$ $\rightarrow$ exact algorithm for MSR in this case?

Theorem

There is a polynomial time exact algorithm for the unweighted MSR problem when no clusters of radius zero) is allowed.

- Euclidean MSD: exact algorithm for constant k. Euclidean MSD with variable k?
- 2-approximation for Euclidean MSD + ratio 2 hardness for general MSD. Can we beat factor 2?

Theorem

There is a PTAS for the Euclidean MSD which runs in $n^{O(1/\epsilon)}$

MSR Restricted to Unweighted Graphs

- Metric: the shortest path metric of an unweighted graph.

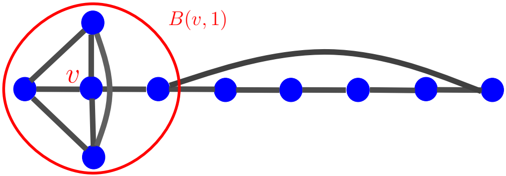
MSR Restricted to Unweighted Graphs

- Metric: the shortest path metric of an unweighted graph.
- Solving MSR for the connected graphs $\rightarrow$ solve the general case.

MSR Restricted to Unweighted Graphs

Definition

$B(v, r)$ the set of vertices $\{u \in V : d(v, u) \leq r\}$

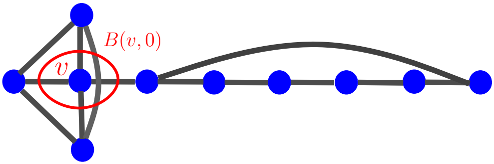


MSR Restricted to Unweighted Graphs

Definition

$B(v, r)$ the set of vertices $\{u \in V : d(v, u) \leq r\}$

zero ball or singleton Ball of radius zero



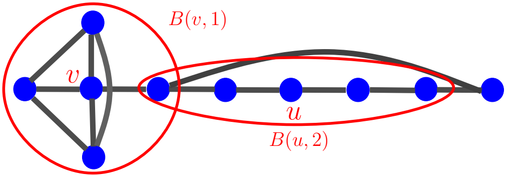
MSR Restricted to Unweighted Graphs

Definition

$B(v,r)$ the set of vertices $\{u \in V : d(v,u) \leq r\}$

zero ball or singleton Ball of radius zero

two balls intersect At least one common vertex



MSR Restricted to Unweighted Graphs

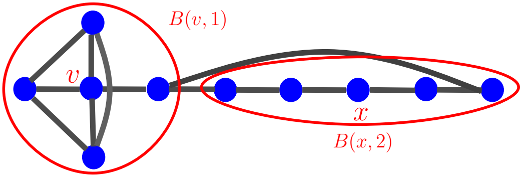
Definition

$B(v, r)$ the set of vertices $\{u \in V : d(v, u) \leq r\}$

zero ball or singleton Ball of radius zero

two balls intersect At least one common vertex

two balls adjacent do not intersect and an edge connecting them



MSR Restricted to Unweighted Graphs

Definition

$B(v, r)$ the set of vertices $\{u \in V : d(v, u) \leq r\}$

zero ball or singleton Ball of radius zero

two balls intersect At least one common vertex

two balls adjacent do not intersect and an edge connecting them

Canonical optimal solution has minimum number of balls

Properties of a canonical optimal solution

Lemma

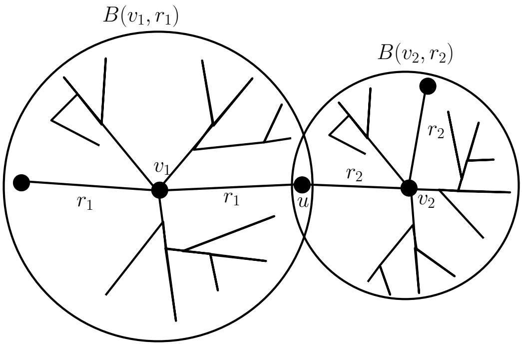
A canonical optimal solution does not have any intersecting balls.

Properties of a canonical optimal solution

Lemma

A canonical optimal solution does not have any intersecting balls.

Proof: Substitute these balls with $B(v, r_1 + r_2)$.

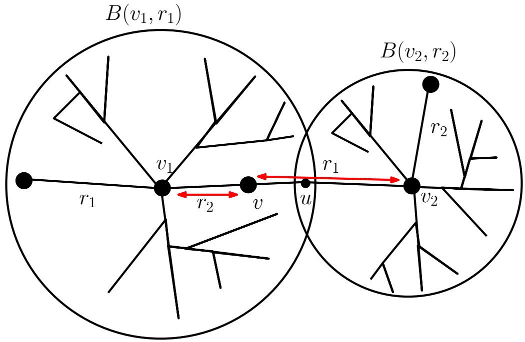


Properties of a canonical optimal solution

Lemma

A canonical optimal solution does not have any intersecting balls.

Proof: Choose v at distance r_2 from v_1 on path v_1–v_2.

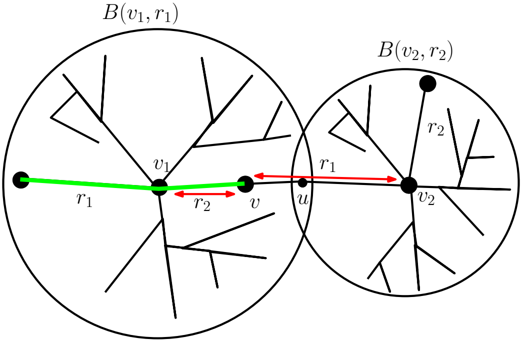


Properties of a canonical optimal solution

Lemma

A canonical optimal solution does not have any intersecting balls.

Proof: Distance of v from $B(v_1, r_1)$ is $r_2 + r_1$.

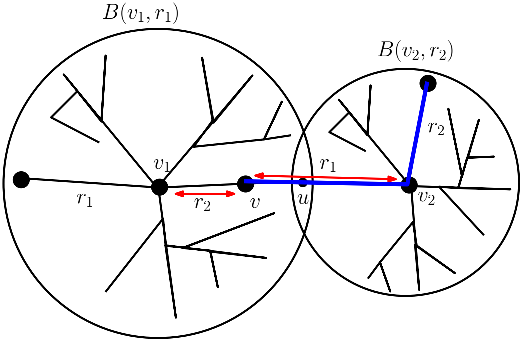


Properties of a canonical optimal solution

Lemma

A canonical optimal solution does not have any intersecting balls.

Proof: Distance of v from $B(v_2, r_2)$ is $r_1 + r_2$.

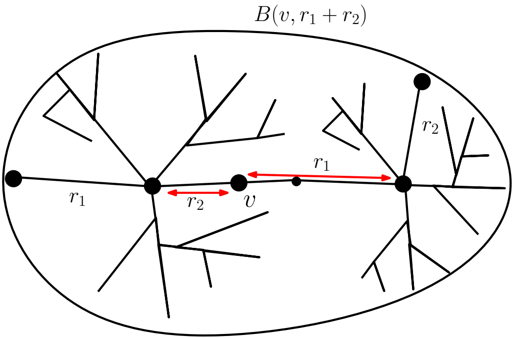


Properties of a canonical optimal solution

Lemma

A canonical optimal solution does not have any intersecting balls.

Proof: Thus, the ball $B(v, r_1 + r_2)$ covers all vertices.



Properties of a canonical optimal solution, Cont'd

Lemma

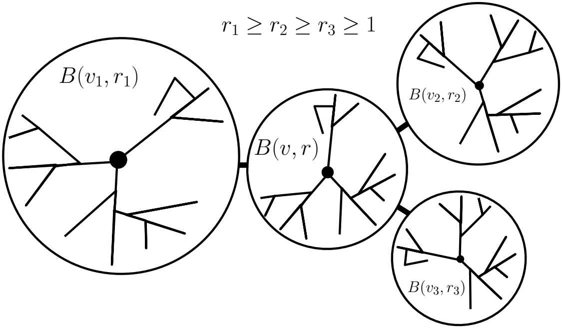
In a canonical optimal solution, each ball is adjacent to at most two balls. (Fails with existence of zero balls.)

Properties of a canonical optimal solution, Cont'd

Lemma

In a canonical optimal solution, each ball is adjacent to at most two balls. (Fails with existence of zero balls.)

Proof: Substitute these balls with $B(u, r + r_1 + r_2 + 1)$.



Properties of a canonical optimal solution, Cont'd

Corollary

In a canonical optimal solution, the balls form a path or cycle.

Properties of a canonical optimal solution, Cont'd

Corollary

In a canonical optimal solution, the balls form a path or cycle.

- Observation: the number of all possible balls $\leq n^2$.

Properties of a canonical optimal solution, Cont'd

Corollary

In a canonical optimal solution, the balls form a path or cycle.

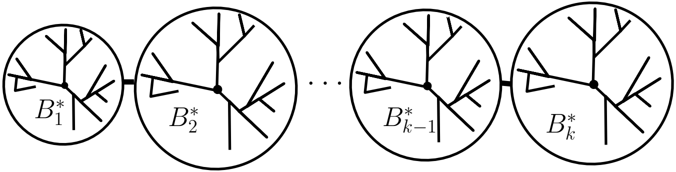
- Observation: the number of all possible balls $\leq n^2$.
- The case of cycle is similar to the case of path.

Properties of a canonical optimal solution, Cont'd

Corollary

In a canonical optimal solution, the balls form a path or cycle.

- Observation: the number of all possible balls $\leq n^2$.
- The case of cycle is similar to the case of path.
- Consider a canonical optimal solution: $B_1^*, B_2^*, \ldots, B_k^*$.



General Idea

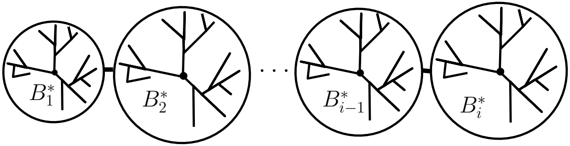
- Guess the last ball, remove it, and solve recursively.

General Idea

- Guess the last ball, remove it, and solve recursively.
- $\textsc{BestCover}(H, j)$:
  1. For all choices of a ball B, $\mathcal{C} \leftarrow B \cup \textsc{BestCover}(H \setminus B, j-1)$.
  2. Return the best solution in $\mathcal{C}$.

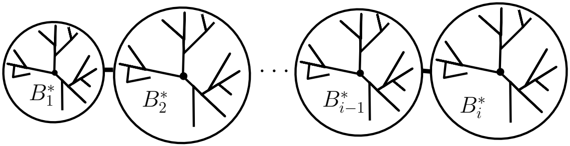
General Idea

- Guess the last ball, remove it, and solve recursively.
- $\textsc{BestCover}(H, j)$:
  1. For all choices of a ball B, $\mathcal{C} \leftarrow B \cup \textsc{BestCover}(H \setminus B, j-1)$.
  2. Return the best solution in $\mathcal{C}$.
- G_i: the first i balls, $G_k = G$ and $G_{i-1} = G_i \setminus B_i^*$



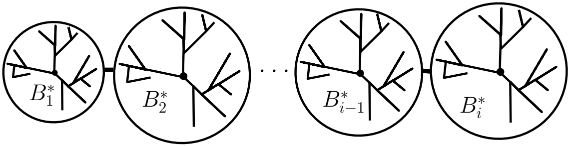
General Idea

- Guess the last ball, remove it, and solve recursively.
- $\textsc{BestCover}(H, j)$:
  1. For all choices of a ball B, $\mathcal{C} \leftarrow B \cup \textsc{BestCover}(H \setminus B, j-1)$.
  2. Return the best solution in $\mathcal{C}$.
- G_i: the first i balls, $G_k = G$ and $G_{i-1} = G_i \setminus B_i^*$
- $\textsc{BestCover}(H, j)$: optimal when $H = G_i$ and $j = i$



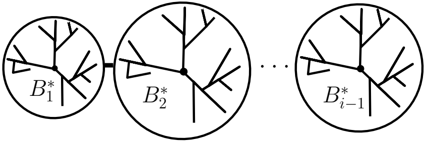
General Idea

- Guess the last ball, remove it, and solve recursively.
- $\textsc{BestCover}(H, j)$:
  1. For all choices of a ball B, $\mathcal{C} \leftarrow B \cup \textsc{BestCover}(H \setminus B, j-1)$.
  2. Return the best solution in $\mathcal{C}$.
- G_i: the first i balls, $G_k = G$ and $G_{i-1} = G_i \setminus B_i^*$
- $\textsc{BestCover}(H, j)$: optimal when $H = G_i$ and $j = i$
- When $H = G_i$, $j = i$, and $B = B_i^*$, $\mathcal{C}$ contains $\textsc{BestCover}(G_{i-1}, i-1) \cup B_i^*$.



General Idea

- Guess the last ball, remove it, and solve recursively.
- $\textsc{BestCover}(H, j)$:
  1. For all choices of a ball B, $\mathcal{C} \leftarrow B \cup \textsc{BestCover}(H \setminus B, j-1)$.
  2. Return the best solution in $\mathcal{C}$.
- G_i: the first i balls, $G_k = G$ and $G_{i-1} = G_i \setminus B_i^*$
- $\textsc{BestCover}(H, j)$: optimal when $H = G_i$ and $j = i$
- When $H = G_i$, $j = i$, and $B = B_i^*$, $\mathcal{C}$ contains $\textsc{BestCover}(G_{i-1}, i-1) \cup B_i^*$.



Dealing with running time

- Without any book keeping, the running time is $O((n^2)^k + k2^n)$.

Dealing with running time

- Without any book keeping, the running time is $O((n^2)^k + k2^n)$.
- Dynamic programming $\textsc{Table}[H, j] \rightarrow O(k2^n)$.

Dealing with running time

- Without any book keeping, the running time is $O((n^2)^k + k2^n)$.
- Dynamic programming $\text{TABLE}[H, j] \to O(k2^n)$.
- **Observation:** we are interested to solve only the subproblems corresponding to graphs G_i.

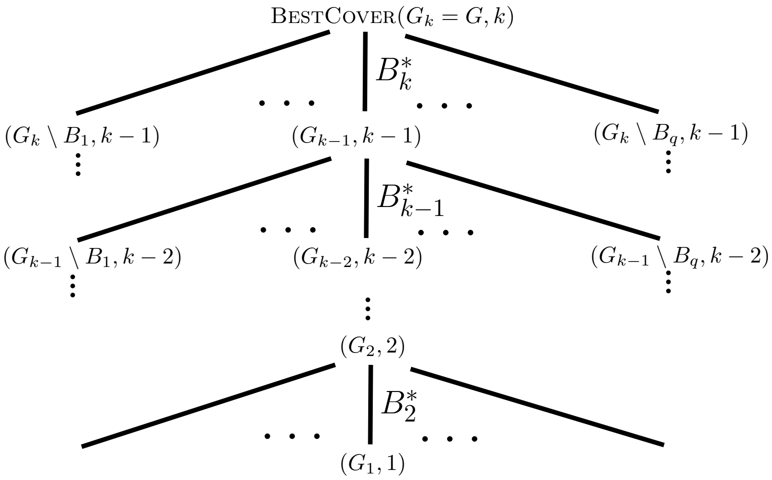
Dealing with running time

- Without any book keeping, the running time is $O((n^2)^k + k2^n)$.
- Dynamic programming $\textsc{Table}[H, j] \to O(k2^n)$.
- **Observation:** we are interested to solve only the subproblems corresponding to graphs G_i.
- $\mathcal{F}$: a poly. size family of subgraphs, contains all G_i.

Dealing with running time

- Without any book keeping, the running time is $O((n^2)^k + k2^n)$.
- Dynamic programming $\textsc{Table}[H, j] \rightarrow O(k2^n)$.
- **Observation:** we are interested to solve only the subproblems corresponding to graphs G_i.
- $\mathcal{F}$: a poly. size family of subgraphs, contains all G_i.
- Run $\textsc{BestCover}(H, j)$ only for $H \in \mathcal{F}$.

Dealing with running time



Finding $\mathcal{F}$

Lemma

$\mathcal{F}$ can be computed in polynomial time, has at most $2n^2 + 1$ members and contains G_i for all $1 \leq i \leq k$.

Finding $\mathcal{F}$

Lemma

$\mathcal{F}$ can be computed in polynomial time, has at most $2n^2 + 1$ members and contains G_i for all $1 \leq i \leq k$.

Proof: The algorithm for finding $\mathcal{F}$ is as follows:

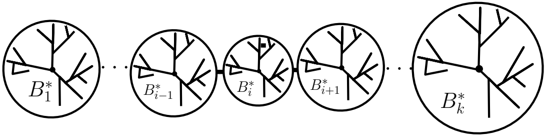
Finding $\mathcal{F}$

Lemma

$\mathcal{F}$ can be computed in polynomial time, has at most $2n^2+1$ members and contains G_i for all $1 \leq i \leq k$.

Proof: The algorithm for finding $\mathcal{F}$ is as follows:

1. For each ball B, consider $G \setminus B$. If it has at most two components, add each component to $\mathcal{F}$.



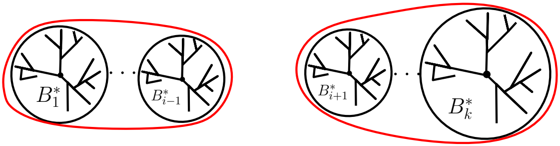
Finding $\mathcal{F}$

Lemma

$\mathcal{F}$ can be computed in polynomial time, has at most $2n^2 + 1$ members and contains G_i for all $1 \leq i \leq k$.

Proof: The algorithm for finding $\mathcal{F}$ is as follows:

1. For each ball B, consider $G \setminus B$. If it has at most two components, add each component to $\mathcal{F}$.



Finding $\mathcal{F}$

Lemma

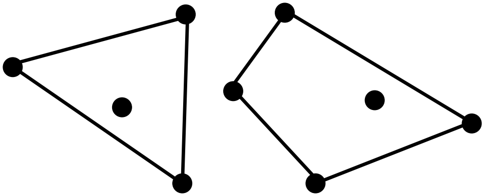
$\mathcal{F}$ can be computed in polynomial time, has at most $2n^2 + 1$ members and contains G_i for all $1 \leq i \leq k$.

Proof: The algorithm for finding $\mathcal{F}$ is as follows:

1. For each ball B, consider $G \setminus B$. If it has at most two components, add each component to $\mathcal{F}$.
2. Add G to $\mathcal{F}$.

Euclidean MSD

- The clusters can be characterized as polygons in plane.



Euclidean MSD

- The clusters can be characterized as polygons in plane.
- Similar Gibson *et al.*'s (SODA'08) exact algorithm for Euclidean MSR.

Euclidean MSD

- The clusters can be characterized as polygons in plane.
- Similar Gibson *et al.*'s (SODA'08) exact algorithm for Euclidean MSR.
- High level idea of Gibson *et al.*'s exact algorithm: separate an instance into two parts by guessing a constant number of discs in optimum.

Euclidean MSD

- The clusters can be characterized as polygons in plane.
- Similar Gibson *et al.*'s (SODA'08) exact algorithm for Euclidean MSR.
- High level idea of Gibson *et al.*'s exact algorithm: separate an instance into two parts by guessing a constant number of discs in optimum.
- The number of possible discs is polynomial $\rightarrow$ one can enumerate all constant size subset of discs in poly. time.

Euclidean MSD

- The clusters can be characterized as polygons in plane.
- Similar Gibson *et al.*'s (SODA'08) exact algorithm for Euclidean MSR.
- High level idea of Gibson *et al.*'s exact algorithm: separate an instance into two parts by guessing a constant number of discs in optimum.
- The number of possible discs is polynomial $\rightarrow$ one can enumerate all constant size subset of discs in poly. time.
- Recursively solve each part using dynamic programming.

Adapting Gibson *et al.*'s Algorithm

Main Difficulties

- Exponential possible clusters.

Adapting Gibson *et al.*'s Algorithm

Main Difficulties

- Exponential possible clusters.
- Thin clusters $\rightarrow$ some packing arguments fails.

Adapting Gibson *et al.*'s Algorithm

Main Difficulties

- Exponential possible clusters.
- Thin clusters $\rightarrow$ some packing arguments fails.
- Our modifications $\rightarrow$ analysis should be changed.

Adapting Gibson *et al.*'s Algorithm

Main Difficulties

- Exponential possible clusters.
- Thin clusters $\rightarrow$ some packing arguments fails.
- Our modifications $\rightarrow$ analysis should be changed.

Handling the first issue

- Approximate each polygon with an enclosing polygon of diameter within factor $(1 + \epsilon)$.

Adapting Gibson *et al.*'s Algorithm

Main Difficulties

- Exponential possible clusters.
- Thin clusters $\rightarrow$ some packing arguments fails.
- Our modifications $\rightarrow$ analysis should be changed.

Handling the first issue

- Approximate each polygon with an enclosing polygon of diameter within factor $(1+\epsilon)$.
- New polygon is simpler: determined by $O(1/\epsilon)$ points $\rightarrow O(n^{1/\epsilon})$ new polygons.

Adapting Gibson *et al.*'s Algorithm

Main Difficulties

- Exponential possible clusters.
- Thin clusters $\rightarrow$ some packing arguments fails.
- Our modifications $\rightarrow$ analysis should be changed.

Handling the first issue

- Approximate each polygon with an enclosing polygon of diameter within factor $(1+\epsilon)$.
- New polygon is simpler: determined by $O(1/\epsilon)$ points $\rightarrow O(n^{1/\epsilon})$ new polygons.
- size c subsets of new polygons, enumerable in $O(n^{\frac{c}{\epsilon}})$.

Adapting Gibson *et al.*'s Algorithm

Main Difficulties

- Exponential possible clusters.
- Thin clusters $\rightarrow$ some packing arguments fails.
- Our modifications $\rightarrow$ analysis should be changed.

Handling the first issue

- Approximate each polygon with an enclosing polygon of diameter within factor $(1+\epsilon)$.
- New polygon is simpler: determined by $O(1/\epsilon)$ points $\rightarrow O(n^{1/\epsilon})$ new polygons.
- size c subsets of new polygons, enumerable in $O(n^{\frac{c}{\epsilon}})$.
- Intuitive Example: A regular polygon and the polygon constructed from extension of every ith edges of it

Conclusion and Future Works

- The difficult core of the problem: finding the zero balls.

Conclusion and Future Works

- The difficult core of the problem: finding the zero balls.
- Open questions: an exact algorithm in presence of singletons? A PTAS for the general version?

Conclusion and Future Works

- The difficult core of the problem: finding the zero balls.
- Open questions: an exact algorithm in presence of singletons? A PTAS for the general version?
- We gave a PTAS for Euclidean MSD. The complexity of Euclidean MSD?

Conclusion and Future Works

- The difficult core of the problem: finding the zero balls.
- Open questions: an exact algorithm in presence of singletons? A PTAS for the general version?
- We gave a PTAS for Euclidean MSD. The complexity of Euclidean MSD?
- The MSD problem with constant k: we found an exact algorithm.

Thanks for your attention!
Questions?