Minimum Shared-Power Edge Cut

Kshitij Jain
University of Waterloo
arxiv/1806.04742

16th June, 2018

Application: Surveillance of US-Mexico border

Barrier Coverage

- A sensor network provides k-barrier coverage, if every path that crosses the width of the belt completely, is covered by at least k distinct sensors.

Barrier Coverage

- A sensor network provides k-barrier coverage, if every path that crosses the width of the belt completely, is covered by at least k distinct sensors.
- How to determine after deploying sensors in a region, whether the region is k-barrier covered?

Barrier Coverage

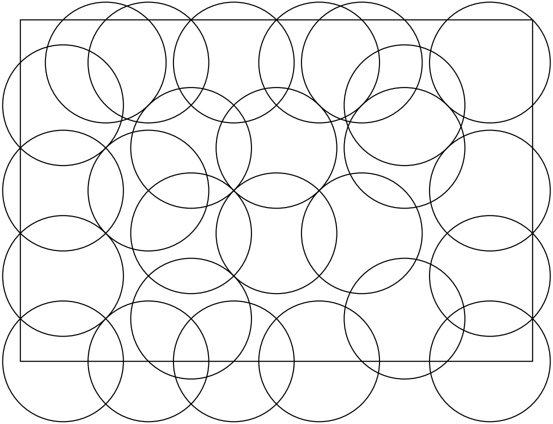


Figure: The rectangular barrier and the disks representing sensors

Barrier Coverage

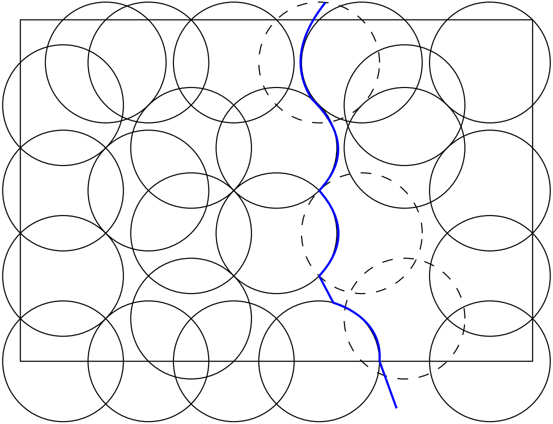


Figure: Path from bottom to top in the free space after removing three of the disks.

Barrier coverage can be computed in polynomial time by applying **Menger's theorem.**

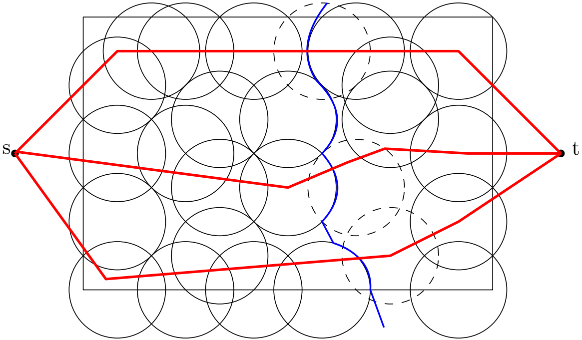
Barrier coverage can be computed in polynomial time by applying **Menger's theorem.**

Menger's Theorem

Let $G = (V, E)$ be a graph and let $S, T \subseteq V$. Then the maximum number of vertex-disjoint S-T paths is equal to the minimum size of an S-T disconnecting vertex set

Solution to Barrier Coverage

Barrier coverage becomes the minimum size of a vertex cut, which is equal to the maximum number of vertex disjoint paths that go from left to right of the rectangle.



Minimum Shrinkage

Definition

The *minimum shrinkage* of the sensor network is the minimum $\sum s_i$ such that if we shrink the i^{th} sensor disk by s_i, then the network no longer provides a barrier i.e., there is a path from bottom to top in the free space between the shrunken disks.

Minimum Shrinkage

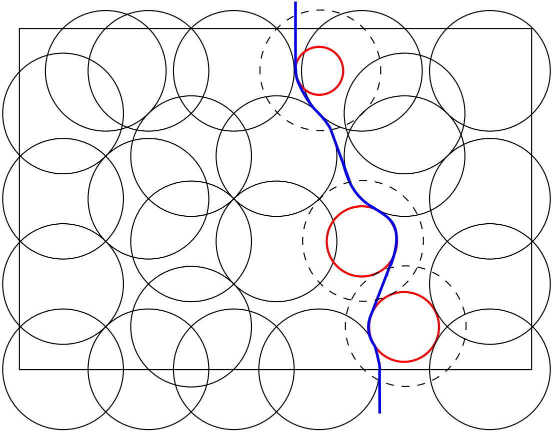


Figure: Path from bottom to top in the free space after shrinking three of the disks.

Minimum Shared-Power Edge Cut

Generalization of minimum shrinkage from unit disc graphs to general graphs:

Minimum Shared-Power Edge Cut (MSPEC)

Input: Graph $G = (V \cup \{s, t\}, E)$ and a non-negative weight $w_{u,v}$ on each edge $(u, v) \in E$.
Problem: Assign a non-negative power p_v to each $v \in V$ so that removing the edge set $\{(u, v) \in E : p_u + p_v \geq w_{u,v}\}$ disconnects s and t and $\sum_{v \in V} p_v$ is minimized.

ILP for MSPEC

$$\min \sum_{u \in V} p_u$$

$$s.t. \sum_{(u,v) \in \pi} x_{u,v} \geq 1 \quad \forall \pi \in \Pi_{st}$$

$$p_u + p_v \geq w_{u,v} x_{u,v} \quad \forall (u,v) \in E$$

$$x_{u,v} \in \{0,1\} \quad \forall (u,v) \in E$$

$$p_u \geq 0 \quad \forall u \in V$$

$$p_s = p_t = 0$$

Integrality Gap

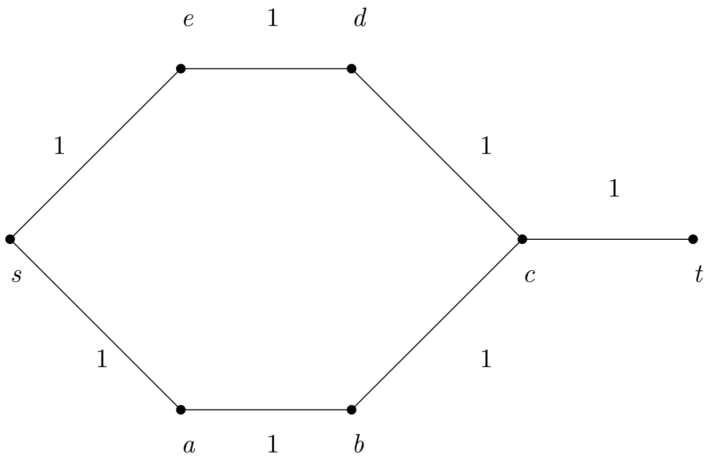


Figure: For this simple unweighted example, integrality gap of the IP is 2. $p_c = 0.5$ and $x_{b,c} = 0.5, x_{c,d} = 0.5$ and $x_{c,t} = 0.5$, is a feasible solution for the LP relaxation of the above ILP.

Alternate forumlation of Minimum Shared-Power Edge Cut

Egerváry's Theorem

Let $G = (V, E)$ be a bipartite graph and let $w : E \rightarrow R_+$ be a weight function. Then the maximum weight of a matching in G is equal to the minimum value of $y(V)$, where $y : V \rightarrow R_+$ is such that $y_u + y_v \geq w_e$ for each edge $e = uv$

Alternate forumlation of Minimum Shared-Power Edge Cut

Egerváry's Theorem

Let $G = (V, E)$ be a bipartite graph and let $w : E \to R_+$ be a weight function. Then the maximum weight of a matching in G is equal to the minimum value of $y(V)$, where $y : V \to R_+$ is such that $y_u + y_v \geq w_e$ for each edge $e = uv$

Minimum Shared-Power Edge Cut (MSPEC)

Given an edge-weighted graph, partition the vertices into two sets with s in one set and t in the other, and minimize the weight of the maximum matching of the edges crossing between the two sets

Minimum Shared-Power Edge Cut

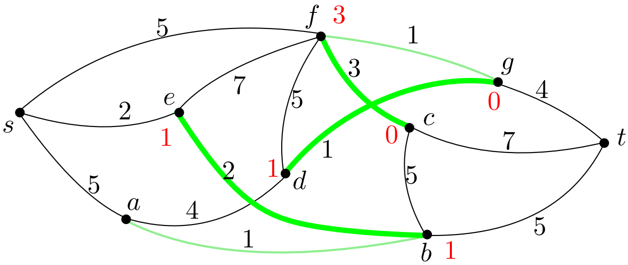


Figure: The powers on the vertices are shown in red. The cut edges are in green and the edges of the maximum matching are shown in bold green.

Minimum Bottleneck Shared-Power Edge Cut

Instead of minimizing the sum, minimize the bottleneck power:

Bottleneck power

Assign the minimum power p^* on every vertex in V such that s and t become disconnected if we remove the edges (u, v) s.t. $w_{u,v} \leq 2p^*$.

Minimum Bottleneck Shared-Power Edge Cut

Instead of minimizing the sum, minimize the bottleneck power:

Bottleneck power

Assign the minimum power p^* on every vertex in V such that s and t become disconnected if we remove the edges (u, v) s.t. $w_{u,v} \leq 2p^*$.

Theorem

Minimum Bottleneck Shared-Power Edge Cut can be solved optimally in $O((n + m) \log m)$time.

Minimum Bottleneck Shared-Power Edge Cut

Instead of minimizing the sum, minimize the bottleneck power:

Bottleneck power

Assign the minimum power p^* on every vertex in V such that s and t become disconnected if we remove the edges (u, v) s.t. $w_{u,v} \leq 2p^*$.

Theorem

Minimum Bottleneck Shared-Power Edge Cut can be solved optimally in $O((n + m) \log m)$time.

Idea: You only need to check if $\mathrm{p}^* = \frac{1}{2} w_{u,v}$ for all edges. Binary search on all such values and check s-t connectivity using BFS/DFS.

This problem infact generalizes a broader class of barrier coverage problems known as the Maximum Breach Path where the optimal solution is computed in similar running time using Voronoi diagrams.

Approximation Scheme

Theorem

There exists a fully polynomial time approximation scheme (FPTAS) for the Minimum Shared-Power Edge Cut problem.

Approximation Scheme

Theorem

There exists a fully polynomial time approximation scheme (FPTAS) for the Minimum Shared-Power Edge Cut problem.

FPTAS

FPTAS for problem X is an approximation scheme whose time complexity is polynomial in the input size and also polynomial in $1/\epsilon$. You get an approximation guarantee of $1 + \epsilon$ with running time inversely proportional to ϵ.

Proof Ideas

- The idea is to convert the MSPEC problem to a minimum vertex cut problem.
- If we can only assign power 0 or 1 to every vertex in V, then MSPEC is equivalent to the Minimum Vertex Cut.

Proof Ideas

- The idea is to convert the MSPEC problem to a minimum vertex cut problem.
- If we can only assign power 0 or 1 to every vertex in V, then MSPEC is equivalent to the Minimum Vertex Cut.
- Discretize MSPEC by replacing each vertex $v \in V$ by multiple copies of v such that removing one copy corresponds to assigning a small fraction of the maximum power to v.

Proof Ideas

- **Aim:** Discretization introduces an error of at most $\alpha = \frac{\varepsilon}{n}$ for each vertex with respect to the OPT.
- Need an upper bound on the maximum power that can be assigned to any vertex, and, for the error analysis, we need a lower bound on the optimum solution.

Proof Ideas

- **Aim:** Discretization introduces an error of at most $\alpha = \frac{\varepsilon}{n}$ for each vertex with respect to the OPT.
- Need an upper bound on the maximum power that can be assigned to any vertex, and, for the error analysis, we need a lower bound on the optimum solution.
- Use minimum bottleneck shared-power edge cut to get these bounds.
- We get this by proving $p^* \leq \text{OPT} \leq np^*$.

Theorem

Theorem

There is a fully polynomial time approximation scheme (FPTAS) for the Minimum Shared-Power Edge Cut problem (MSPEC) with the running time of $O(n^{5.5} m \varepsilon^{-2.5})$.

- The running time is dominated by finding the min-vertex cut in the discretized graph. A min vertex cut can be found in $O(n^{1/2} m)$ time.

Theorem

Theorem

There is a fully polynomial time approximation scheme (FPTAS) for the Minimum Shared-Power Edge Cut problem (MSPEC) with the running time of $O(n^{5.5} m \varepsilon^{-2.5})$.

- The running time is dominated by finding the min-vertex cut in the discretized graph. A min vertex cut can be found in $O(n^{1/2} m)$ time.
- If our original graph G has n vertices and m edges, then the discretized graph G' has $|V'| = O(\frac{n^3}{\varepsilon})$ vertices and at most $|E'| = O(\frac{mn^4}{\varepsilon^2})$ edges.

Theorem

Theorem

There is a fully polynomial time approximation scheme (FPTAS) for the Minimum Shared-Power Edge Cut problem (MSPEC) with the running time of $O(n^{5.5} m \varepsilon^{-2.5})$.

- The running time is dominated by finding the min-vertex cut in the discretized graph. A min vertex cut can be found in $O(n^{1/2} m)$ time.
- If our original graph G has n vertices and m edges, then the discretized graph G' has $|V'| = O(\frac{n^3}{\varepsilon})$ vertices and at most $|E'| = O(\frac{mn^4}{\varepsilon^2})$ edges.
- Combining this we get a running time of $O(n^{5.5} m \varepsilon^{-2.5})$.

Speedup

Instead of using Minimum Bottleneck Shared-Power Edge Cut for the discretization purpose, we use a 2-approximate solution of MSPEC for the discretization.
Let Z be the approximate solution then, we get $Z \leq \text{OPT} \leq 2Z$

Speedup

Instead of using Minimum Bottleneck Shared-Power Edge Cut for the discretization purpose, we use a 2-approximate solution of MSPEC for the discretization.

Let Z be the approximate solution then, we get $Z \leq \text{OPT} \leq 2Z$

Faster FPTAS

An improved running time of $O(n^3 m \epsilon^{-2.5})$.

Open Questions

The biggest open question is to settle the complexity of the **Minimum Shared-Power Edge Cut**—is it in P? NP-hard?

Open Questions

The biggest open question is to settle the complexity of the **Minimum Shared-Power Edge Cut**—is it in P? NP-hard?

Is it easier to solve the the minimum shrinkage problem? Develop a polynomial time algorithm for it?

Questions?