Minimum Shared-Power Edge Cut

by

Kshitij Jain

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Mathematics
in
Computer Science

Waterloo, Ontario, Canada, 2018

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

I would like to acknowledge the names of my co-authors who contributed to the research described in this dissertation, these include:

- Prof. Sergio Cabello
- Prof. Anna Lubiw
- Prof. Debajyoti Mondal

A preprint of this work is available online on the arxiv [8].

Abstract

We introduce a problem called the Minimum Shared-Power Edge Cut (*MSPEC*). The input to the problem is an undirected edge-weighted graph with distinguished vertices s and t, and the goal is to find an s-t cut by assigning "powers" at the vertices and removing an edge if the sum of the powers at its endpoints is at least its weight. The objective is to minimize the sum of the assigned powers.

MSPEC is a graph generalization of a barrier coverage problem in a wireless sensor network: given a set of unit disks with centers in a rectangle, what is the minimum total amount by which we must shrink the disks to permit an intruder to cross the rectangle undetected, i.e. without entering any disc. This is a more sophisticated measure of barrier coverage than the minimum number of disks whose removal breaks the barrier.

We develop a fully polynomial time approximation scheme (FPTAS) for *MSPEC*. We give polynomial time algorithms for the special cases where the edge weights are uniform, or the power values are restricted to a bounded set.

Although *MSPEC* is related to network flow and matching problems, its computational complexity (in P or NP-hard) remains open.

Acknowledgements

This thesis would have been impossible without the support of my supervisor, Anna Lubiw. She provided me with ample guidance, feedback, and advice during my Masters. I have learned a lot under her supervision. I am thankful to her not only for the thesis but also for making my time memorable at the University of Waterloo.

I would like to thanks my readers Therese Biedl, and Lap Chi Lau. Both had a strong influence on my research. I have enjoyed collaborating with Therese during the Algorithms & Complexity group problem sessions. I would also like to thank Lap Chi for his helpful suggestions.

I would like to thank Debajyoti Mondal for useful discussions during our weekly meetings. I would also like to thank Vijay Menon for helpful discussions and being a patient listener.

I thank my parents and my sister for their incredible support during my Masters. I would also like to thank my friends and family members back in India, who made a sincere effort to make sure that I feel homely. My journey would have been impossible without the support of my friends at Waterloo.

Dedication

This is dedicated to my parents and Kavya.

Table of Contents

List of Figures

Chapter 1

Introduction

Minimum weight edge cuts in graphs are very well-studied. In this thesis we look at a variation that arises from unit disk graphs, i.e., intersection graph of unit disks, and other situations where the edges of the graph are determined by geometric properties of the vertices. In this variation, we assign a *power* p_v to each vertex v and an edge $e = (u, v)$ is removed if $p_u + p_v$ is at least the weight of edge e. The goal is to remove the edges of an s-t cut while minimizing the power sum. More formally, the *Minimum Shared-Power Edge Cut* (*MSPEC*) problem is defined as follows:

Input: Graph $G = (V \cup \{s, t\}, E)$ with n vertices, m edges, 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 $v \in V$ and assign $p_s = p_t = 0$ 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.

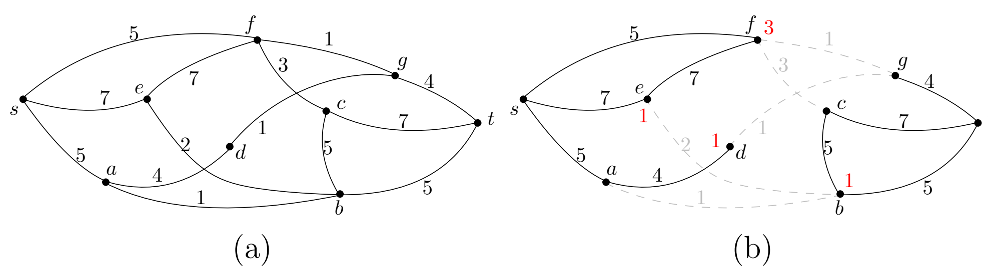


Figure 1.1: Minimum Shared-Power Edge Cut problem: (a) shows the input edge-weighted graph G; (b) shows a s-t cut with the cut edges in gray and the assigned power value in red.

There has been substantial work in the area of activation and survivable network design. Motivated by the application in wireless networks, survivable network design involves designing minimum cost networks subject to connectivity requirements. The cost involves an activation cost to activate the set of edges. The cost function could depend on the edge weights, vertex weights or a combination of the two. We describe activation network design in detail in Section 1.2.

The main result of this thesis is a fully polynomial time approximation scheme (FPTAS) for the minimum shared-power edge cut problem.

A special case of *MSPEC*—and our original motivation for studying it—is the problem of measuring barrier coverage of a sensor network. A sensor network is typically modeled as a collection of unit disks in the plane, where each disk represents the sensing region of its corresponding sensor [45]. Such a collection of unit disks in the plane is represented by a *unit disk graph*. Given a set P of points in $\mathbb{R}^2$, the *unit disk graph,* $UDG(P)$, is defined by setting its vertex set to P (in particular, set P corresponds to the centers of unit disks) and creating an edge between any two points of P whose Euclidean distance is at most two. Figure 1.2 shows a rectangular barrier and unit disks representing sensors.

The network provides a *barrier* between regions R_1 and R_2 if every path from R_1 to R_2 intersects the union of the disks. One simple quantitative measure of barrier strength is the minimum number of disks whose removal permits a path from R_1 to R_2 in the free space outside the disks. This can be computed in polynomial time [25] for the special case of a *rectangular barrier*, where the sensors lie in a rectangle and must block paths from the bottom to the top of the rectangle. Figure 1.3 shows the optimal solution for this measure for the sensor network modeled as in Figure 1.2.

We suggest, as a more sophisticated measure, the minimum total amount by which we must shrink the disks to permit such a path. This measure reflects the reality that sensor strength typically deteriorates ("attenuates") with distance from the sensor [45]. For a rectangular barrier, our *Minimum Shrinkage* problem (as in Figure 1.4) can be modeled as *MSPEC*, where we have a vertex for each disk, and the power assigned to a vertex tells us how much to shrink the corresponding disk. We explain barrier coverage and minimum shrinkage in detail in Chapter 2. Thus the main result of this work provides an approximation scheme to compute Minimum Shrinkage for rectangular barrier coverage.

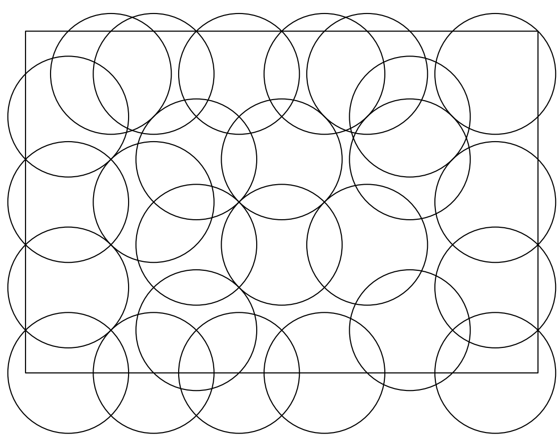


Figure 1.2: Shows the rectangular barrier and unit disks representing sensors.

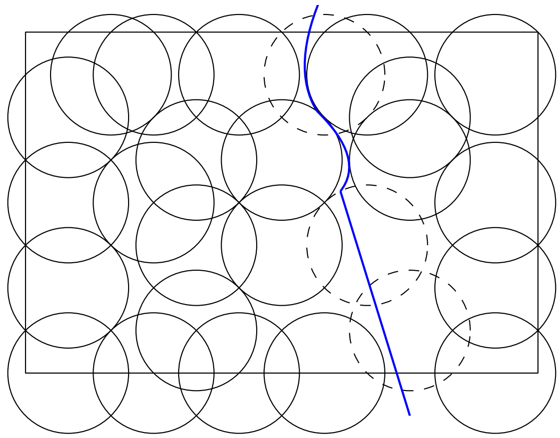


Figure 1.3: Path from bottom to top in the free space after removing three of the disks.

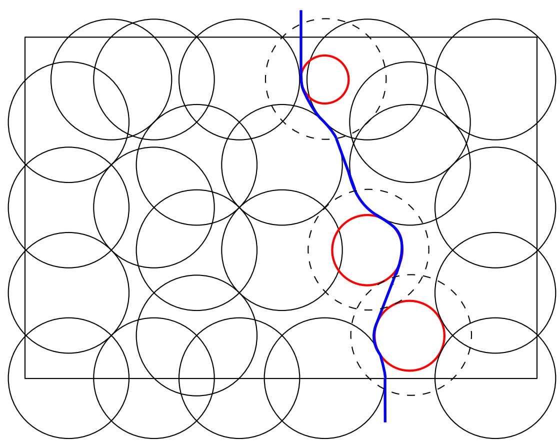


Figure 1.4: Minimum shrinkage for the rectangular barrier coverage problem: shows a path from bottom to top in the free space after shrinking three of the disks.

Outline of the Thesis

In Section 1.1 of this chapter, we explain the preliminaries that will help readers to understand the thesis better. In Section 1.2, we give a brief survey of the related models and problems considered in the literature. We conclude this chapter by giving alternative formulations of *MSPEC*. In Chapter 2, we give a survey on barrier coverage and formally introduce minimum shrinkage. We show the connection between minimum shrinkage and *MSPEC*. Chapter 3 is the heart of the thesis where we describe our approximation algorithm for *MSPEC* and also give a polynomial time algorithm for the bottleneck version of *MSPEC*. In Chapter 4, we discuss some possible variants of *MSPEC*. In Chapter 5, we show to speed up our approximation algorithm. Chapter 6 concludes the thesis and suggests directions for future work.

1.1 Preliminaries

In this section we give the basic definitions and theorems that will assist in understanding the problem formulation and proofs.

1.1.1 FPTAS

A *Fully Polynomial Time Approximation Scheme, FPTAS*, is an $1+\epsilon$ approximation scheme whose running time is polynomial in the input size and also polynomial in $1/\epsilon$ (for instance, a running time of $O((n/\epsilon)^c)$) [43].

In contrast, a *polynomial time approximation scheme, PTAS*, is an $1+\epsilon$ approximation scheme whose running time is only polynomial in the input size (for instance, a running time of $O(n^{1/\epsilon})$).

1.1.2 Menger's Theorem

Let $G = (V, E)$ be a graph and let $S, T \subseteq V$. A path is an S-T *path* if it runs from a vertex in S to a vertex in T. A set $C \subseteq V$ of vertices is called S-T *disconnecting* if C intersects each S-T path.

Theorem 1 (Menger's theorem, e.g. [39, chapter 9])**.** *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.*

1.1.3 Kőnig's Theorem

Theorem 2 (Kőnig's theorem, see [38] or [39, chapter 16])**.** *For any bipartite graph $G = (V, E)$, the maximum size of a matching is equal to the minimum size of a vertex cover.*

For a general graph G, this relationship is an inequality instead of an equality, i.e., the maximum size of a matching is at most the minimum size of a vertex cover.

1.1.4 w-vertex cover

We will need a generalization of Kőnig's theorem (Theorem 2) to the case of edge weights, which was proved by Egerváry.

Let $G = (V, E)$ be a graph and let $w : E \to \mathbb{R}^+$. A *w-vertex cover* is a vector $y : V \to \mathbb{R}^+$ such that $y_u + y_v \geq w_e$ for each edge $e = (u, v)$ [39, chapter 17].

In general computing a minimum w-vertex cover is NP-hard since it generalizes the vertex cover problem. But for bipartite graphs, the minimum w-vertex cover can be computed in polynomial time. This is a consequence of Egerváry's characterization of the maximum weight matching for bipartite graphs.

Theorem 3 (Egerváry's theorem, see [24] or [39, chapter 17]). *Let $G = (V, E)$ be a bipartite graph and let $w : E \to \mathbb{R}^+$ be a weight function. Then the maximum weight of a matching in G is equal to the minimum value of $y(V)$, where $y : V \to \mathbb{R}^+$ is such that $y_u + y_v \geq w_e$ for each edge $e = (u, v)$.*

Using the *Hungarian method* [23] the maximum-weight bipartite matching can be found with running time $O(n(m + n \log n))$. As a consequence of Theorem 3, it is also possible to find the minimum w-vertex cover from the given maximum-weight bipartite matching with running time $O(n(m + n \log n))$ [19].

1.1.5 Total Unimodularity

A more general tool that implies Egervary's theorem (Theorem 3), and many other min-max equalities, is the theory of total unimodularity. Total unimodularity of matrices is an important tool in the area of integer programming. A matrix A is called *totally unimodular* if each square submatrix of A has a determinant either $-1, 0$, or $+1$.

Theorem 4 (Hoffman-Kruskal Theorem, see [18] or [39, chapter 5]). *A is a totally unimodular (TUM) $m \times n$ matrix if and only if for all integral $b \in \mathbb{Z}^m$, the polyhedron $P = \{x | Ax \leq b\}$ is an integral polyhedron.*

A polyhedron P is called an *integer polyhedron* if it is the convex hull of the integer vectors contained in P. This gives us a strong property that the linear program solution will be equal to the integer program solution in such cases i.e., $\max\{c^T x \mid Ax \leq b; x \text{ integer}\}$ $= \max\{c^T x \mid Ax \leq b\}$.

Theorem 5 (See [39, Theorem 18.2]). *A graph $G = (V, E)$ is bipartite if and only if its "vertex-edge incidence matrix" A is totally unimodular.*

1.2 Related Models

Motivated by problems in design of wireless networks, there is substantial previous work on graph optimization problems in which edges are selected depending on "power" values p_v that are assigned to each vertex v. The objective is to minimize the sum of the powers, while satisfying connectivity properties for the selected edges that are of interest in the design of ad-hoc wireless networks. For example, the selected edges form a spanning tree, or a k-connected subgraph.

For edge-weighted graphs the models that have been considered for selecting, or "activating", an edge (u, v) of weight $w_{u,v}$ are:

1. $\min\{p_u, p_v\} \geq w_{u,v}$ ("Power Optimization", e.g. [15])
2. $\max\{p_u, p_v\} \geq w_{u,v}$ ("Max Power Optimization", e.g. [3])
3. $p_u + p_v \geq w_{u,v}$ ("Installation Cost Optimization", e.g. [34])
4. a more general function of p_u and p_v ("Activation Networks", e.g. [34])

In general, in the activation network problems we are given a graph $G = (V, E)$, a domain, D, of parameter values, activation function $f_{u,v} : D \times D \to \{0, 1\}$ for every edge $(u, v) \in E$, and a connectivity requirement C. The goal is to select $p_v \in D$ at each vertex v such that it minimizes the sum $\sum_v p_v$ and the set of activated edges satisfies the connectivity requirement C [35].

There is also an older notion of Node-Weighted Optimization, e.g. [20], for vertex-weighted graphs. For vertices u and v with weight w_u and w_v, respectively, the requirement to activate the edge (u, v) is that $p_u \geq w_u$ and $p_v \geq w_v$.

1.2.1 Connectivity Requirements

In the literature there are a lot of connectivity requirement problems considered for activation network design. Some of the important ones are:

1. **Minimum Vertex-connected Activation Network**: Each pair of vertices should have k vertex-disjoint paths between them in the activated subgraph, for some given $k > 0$.
2. **Minimum Edge-connected Activation Network**: Each pair of vertices should have k edge-disjoint paths between them in the activated subgraph, for some given $k > 0$.
3. **Minimum Steiner Activation Network**: Each pair of vertices in each of k terminal sets $R_1, R_2, \ldots, R_k \subseteq V$ should be connected in the activated subgraph.
4. **Minimum Spanning Activation Tree**: The activated set of edges must contain a spanning tree on all vertices $v \in V$.
5. **Minimum Activation Path**: The activated set of edges must contain a path between two specified vertices $\{s, t\} \in V$.

1.2.2 Known results for power optimization

In the most well-studied model, called "Power Optimization," an edge (u, v) with weight $w_{u,v}$ is selected, or "activated", if p_u and p_v are both at least $w_{u,v}$ i.e., $\min\{p_u, p_v\} \geq w_{u,v}$.

The power at a vertex will be equal to the weight of one of its incident edges, so the possible powers form a discrete set.

Calinescu and Wan [9] gave a $2k$-approximation algorithm for minimum edge-connected activation network. Hajiaghayi et al. [15] gave an $O(\sqrt{n})$-approximation algorithm for the same. Lando and Nutov [26] improved the approximation ratio to k. They showed it cannot be approximated within a factor of $o(\log n)$ unless P=NP. Panigrahi [34] gave a bi-criteria approximation algorithm with the approximation factor of $O(\log n \log k, 2)$. Here, $O(\log n \log k, 2)$ approximation means that his result achieve an approximation factor of $O(\log n \log k)$ while achieving an edge connectivity of $k/2$ instead of k. Note that Panigrahi's result extends to the installation cost framework also.

Hajiaghayi et al. [15] observed that the directed version of the minimum vertex-connected activation network is in P, whereas the directed version of the minimum edge-connected activation network cannot be approximated within a factor of $O(2^{\log^{1-\epsilon} n})$ for any fixed $\epsilon > 0$, unless NP-hard problems can be solved in quasi-polynomial time.

For k=2, Kortsarz and Nutov [22] gave an 11/3-approximation algorithm for the minimum vertex-connected activation network. For the minimum edge-connected activation network, Calinescu et al. [9] gave a 4-approximation algorithm for $k = 2$.

Althaus et al. [2] gave a 5/3-approximation algorithm for the minimum spanning activation tree problem. Kortsarz and Nutov [22] gave a 4-approximation algorithm for the minimum Steiner activation network problem.

1.2.3 Known results for max power optimization

The max power optimization model was introduced by Angel et al. [3] where they defined an edge (u, v) to be activated if $\max\{p_u, p_v\} \geq w_{u,v}$. They gave 2-approximation algorithm for the minimum max power vertex cover problem and as a consequence of total unimodularity (Subsection 1.1.5), they have a polynomial time algorithm for bipartite graphs. They also showed that under this model minimum max power s-t cut (in the paper they call it Min-Power-Cover Cut) can be solved in polynomial time. The key idea here is that the optimal power at a vertex will be equal to the weight of one of its incident edges, so there are only a discrete set of possible powers that may be assigned to a node. Using this property, they create a modified graph and solve the traditional min s-t cut problem on this modified graph to get an optimal answer for the max power s-t cut. They also showed that the minimum spanning activation tree problem in this model is as hard to approximate as the dominating set problem.

1.2.4 Known results for installation cost optimization

In the "installation cost" model, an edge (u, v) with weight $w_{u,v}$ is selected if $p_u + p_v \geq w_{u,v}$.

Panigrahi [34] gave an $O(\log n)$ approximation algorithm for the minimum spanning activation tree problem. He proved that this approximation factor is best possible assuming P $\neq$ NP [34]. Since minimum spanning tree is a special case of minimum vertex-connected activation network, minimum edge-connected activation network, and minimum Steiner activation network with $k = 1$, it is also NP-hard to approximate these problems to a factor of $o(\log n)$.

1.2.5 Known results for activation network

The unified model of activating edges was introduced by Panigrahi [34]. In the activation network model, there is an activation function $f_{u,v} : D \times D \to \{0, 1\}$ that takes as input the values of x_u and x_v and returns 1 if the edge (u, v) is activated for the chosen values of x_u and x_v. In this setting there are generally two assumptions made:

- **Finite domain.** It is assumed that power values are chosen from a finite set of discrete values D and therefore the running time of any algorithm can be polynomial in terms of the size of D.
- **Monotonicity.** It is assumed that the activation function is monotonic, i.e., if $f_{x_u,x_v} = 1$ then $f_{y_u,y_v} = 1$ for any y_u and y_v satisfying $y_u \geq x_u$ and $y_v \geq x_v$.

In this setting Panigrahi [34] showed that it is NP-hard to approximate minimum spanning tree to a factor of $o(\log n)$ and gave a greedy algorithm with an approximation factor of $O(\log n)$. For $k = 1$ and 2, he also gave $O(\log n)$-approximation algorithms for both minimum vertex-connected activation network and minimum edge-connected activation network problems. He also showed that even for $k = 1$, it is NP-hard to obtain an approximation factor of $o(\log n)$ for both problems. He gave a polynomial time algorithm to solve the minimum activation path problem.

1.2.6 Known results for node-weighted optimization

The node-weighted optimization model is slightly different from all other models as the cost is associated with the nodes instead of the edges. The goal is to minimize the sum of the assigned powers p_v on vertices $v \in V$ i.e., $\min \sum_{v \in V} p_v$. An edge (u, v) is said to be activated if $p_u \geq w_u$ and $p_v \geq w_v$. In this model, Klein and Ravi [20] gave an $O(\log n)$ approximation algorithm for the minimum node-weighted Steiner activation network for k=1. In the same paper they also mentioned that approximating minimum node-weighted Steiner tree is as hard as set cover and therefore unless $P = NP$, this is the best approximation factor.

For further background on activation network design problems, we refer readers to Nutov's survey [33].

1.3 Alternative Formulations of *MSPEC*

In this section we give two alternative formulations of the *MSPEC* problem.

1.3.1 *MSPEC* as a Matching

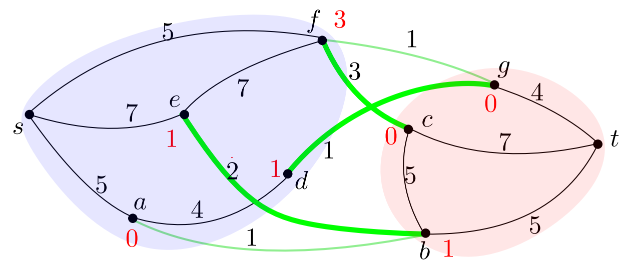
In *MSPEC* we are searching for an edge cut. The set of edges in an s-t cut form a bipartite graph. Given an edge cut C, the power assignment that will remove those edges corresponds to a minimum "w-vertex cover"(see Subsection 1.1.4), i.e., an assignment of weights y_v to the vertices so that $y_u + y_v \geq w_{u,v}$ for all edges (u, v) in the cut C. In a bipartite graph, due to Theorem 3, we know that the minimum value of a w-vertex cover is equal to the maximum weight matching. Therefore, *MSPEC* can be alternatively stated as: 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 a maximum matching of the edges crossing between the two sets, e.g., see Figure 1.5(a).

1.3.2 Integer Program formulation for *MSPEC*

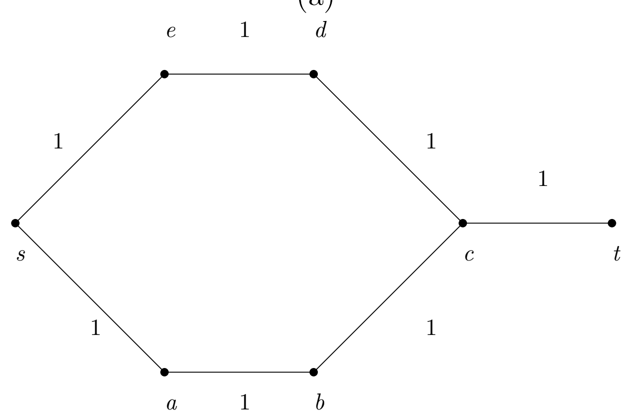
We can also formulate *MSPEC* as an integer program (ILP). Although we do not use it in our algorithms, it may lead to future developments. Let Π_{st} be the set of all paths in G from s to t. We will have a non-negative variable p_u for each $u \in V$ and a 0-1 variable $x_{u,v}$ for each edge $(u, v) \in E$ with the intended interpretation that the cut edges have $x_{u,v} = 1$.

$$
\begin{aligned}
&\min \sum_{u \in V} p_u \\
&s.t. \sum_{(u,v) \in \pi} x_{u,v} \geq 1 && \forall \pi \in \Pi_{st} \\
&p_u + p_v \geq w_{u,v} x_{u,v} && \forall (u, v) \in E \\
&x_{u,v} \in \{0, 1\} && \forall (u, v) \in E \\
&p_u \geq 0 && \forall u \in V \\
&p_s = p_t = 0
\end{aligned}
\tag{1.1}
$$

This integer programming formulation has an integrality gap of at least 2 even for unweighted graphs. Consider the graph shown in Figure 1.5(b). It is easy to verify that the optimum integral solution has cost 1. On the other hand, if we assign $p_c = 0.5$ and $x_{b,c} = 0.5, x_{c,d} = 0.5$ and $x_{c,t} = 0.5$, this is a feasible solution for the LP relaxation of the above ILP. The integrality gap of this ILP is thus $\geq \frac{1}{0.5} = 2$.



(a)



(b)

Figure 1.5: (a) Illustration for *MSPEC*. 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. (b)The integrality gap of ILP (1.1) is 2 for this graph.

Chapter 2

Barrier Coverage in a Sensor Network

In this chapter, we discuss the motivation for studying *MSPEC* which arises from the problem of barrier coverage. We discuss related work in the area of barrier coverage. In Section 2.2, we propose a new measure of barrier coverage, i.e., "Minimum Shrinkage". In Section 2.2.1 we give a reduction from the minimum shrinkage problem to *MSPEC*.

Our motivation for studying the Minimum Shared-Power Edge Cut problem comes from a problem of measuring barrier coverage in a sensor network. A sensor network consists of a set of sensors, where each sensor is modelled as a unit disk (a disk of radius 1) in the plane. The location of a sensor is the center point of its unit disk, and the sensor detects points within its unit disk.

A sensor network provides a barrier between one region of the plane, R_1, and another region, R_2, if any point that travels from R_1 to R_2 will be detected by the sensors, i.e., there is no path between the two regions that remains outside all the disks.

In a *rectangular barrier* the sensors are located at points in a rectangle and the sensor network must detect any path that crosses the rectangle from below to above. Kumar et al. [25] introduced this concept and suggested measuring "barrier coverage" as the minimum value k such that every path that crosses the barrier intersects at least k sensor disks. Equivalently, the barrier coverage is the minimum number of sensors that must be removed to allow a path in the free space between disks. For a rectangular barrier, Kumar et al. showed that barrier coverage can be computed in polynomial time by applying Menger's theorem (Theorem 1) to the intersection graph of the disks—barrier coverage becomes the minimum size of a vertex cut, which is equal to the maximum number of vertex disjoint paths that go from the left of the rectangle to the right of the rectangle. Figure 2.1 shows the relationship between vertex disjoint paths and the minimum barrier coverage.

A number of alternative measures of barrier coverage have been proposed in the literature with the (conflicting) goals of having a measure that models reality and that can be

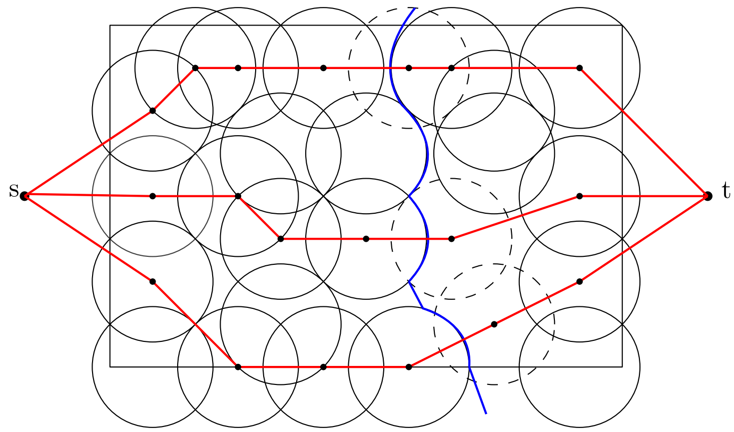


Figure 2.1: Barrier coverage is equal to the maximum number of vertex disjoint paths that go from left to right of the rectangle

computed efficiently. We propose a new measure called Minimum Shrinkage. Like some of the previous measures (see the following section for more background) it models the fact that a sensor's ability to detect points decreases with distance from the sensor. Our approximation algorithm for Maximum Shared-Power Edge Cut can be applied to compute Minimum Shrinkage for rectangular barrier coverage.

Our new measure is defined in Section 2.2, and the reduction to *MSPEC* is given in Section 2.2.1. We begin with further background on barrier coverage in Section 2.1.

2.1 Background on Barrier Coverage

As mentioned above, Kumar et al. [25] introduced the idea of measuring barrier coverage by the minimum number of sensors whose removal permits a path from region R_1 to region R_2 in the free space outside the sensor disks.

Bereg and Kirkpatrick [6] applied the measure of Kumar et al. [25] more broadly and called it "resilience". They introduced a new measure of barrier coverage called *thickness* which is defined to be the minimum, over all paths from R_1 to R_2, of the number of times the path enters a disk, counting repeats. Whereas resilience seems to be hard to compute except for the case of a rectangular barrier, thickness can easily be computed in polynomial time using weighted shortest path algorithms. Although a path from R_1 to R_2 that intersects at most k discs may intersect each of those discs many times, Bereg and Kirkpatrick show that a Euclidean shortest path that intersects only those k discs will intersect any one disc at most three times. Using this structural property, they showed that thickness provides a 3-approximation to resilience. The approximation factor can be improved to 2 if R_1 and R_2 are well separated.

Korman et al. [21] showed that computing resilience of bounded fat regions is NP-hard.

They also gave a fixed-parameter tractable algorithm (parameterized by the resilience) for unit disks.

The above measures implicitly assume that a sensor can detect points uniformly across its unit disk. In reality, the power of a sensor decreases with distance, and a sensor can detect closer points more easily than farther points. In order to take distance from the sensors into account, we can think of diminishing the power of the sensors, or shrinking their unit disks. Meguerdichian et al. [29] formulated this as a "maximum breach path"—to find the maximum value d such that there is a path from region R_1 to R_2 where every point of the path is at least distance d from every sensor point. They showed that a maximum breach path should travel on the Voronoi diagram of the sensor points, and thus can be computed in time $O(n \log n)$. The maximum breach measure is equivalent to asking for the maximum value d such that shrinking all the sensor disks to radius d permits a path in the free space between the shrunken disks.

In a related paper Megerian et al. [28] introduced the notion of "exposure". This measure takes into account, not only how close the path goes to the sensor points, but also the amount of time that is spent close to sensor points. More formally, the "[all] sensor field intensity" at a point is the sum of all the sensor's powers at that point, and "exposure" along a path is the integral of the sensor field intensity along the path. Computing a path of minimum exposure is a very difficult continuous problem. Djidjev [11] gave an approximation algorithm, using the idea of discretizing the domain. For the special case of a rectangular barrier, there is a min-max formula that relates the minimum exposure of a path from bottom to top of the rectangle and a maximum flow from left to right of the rectangle—see Strang [40] and Mitchell [30].

Veltri et al. [44] studied the maximum exposure path as a variant of the minimum exposure path. Given a sensor network with an exposure model, the maximum exposure path between the starting region R_1 and an ending region R_2 is a path connecting the two regions such that this path has the highest exposure among all the R_1-R_2 paths. Intuitively, it can be thought of as the best-case coverage path. They showed computing the maximum exposure path is NP-hard.

For further background on exposure and maximum breach path refer to Ghosh and Das's survey on coverage and connectivity issues in wireless sensor networks [12] (in particular, Section 4).

We discuss related work in the area of barrier coverage which instead of focusing on the worst case behavior, focuses on the expected barrier coverage. Liu et al. [27] showed the necessary condition for barrier coverage under Poisson distribution of the sensors in a 2-D rectangle. They showed that if the width of a rectangle is $\Omega(\log(\mathit{height}))$ and sensors are deployed according to a Poisson process then with high probability, the network is k-barrier covered. A sensor network is *k-barrier covered* if all paths from region R_1 to R_2 are k-covered, i.e., each path intersects with at least k disks.

There has been a long line of study from barrier coverage of non-disk sensor networks to minimizing the movements of sensors for achieving barrier coverage. In the remainder of this section, we discuss other works that have been motivated by barrier coverage.

Gibson et al. [14] studied the opposite problem of minimum barrier coverage, "Isolating points using unit disks". The input is a set of unit disks $\mathcal{D}$, and a set of points $\mathcal{P}$ such that $\mathcal{D}$ *separates* any two points in the $\mathcal{P}$, that is, for any two points p, q in $\mathcal{P}$, every path between p and q intersects at least one disk in D. The goal is to find a minimum cardinality subset of $\mathcal{D}$ that separates $\mathcal{P}$. They gave an $O(1)$ approximation algorithm for this problem. Later Penninger and Vigan [36] showed that this problem is NP-hard. In the journal version of the paper, Gibson et al. [13] improved the approximation ratio to $(9 + \epsilon)$.

Cabello and Giannopoulos [7] considered the above problem for general curves but for two specified points. They formulated "2-point separation" problem as given a set C of n curves in the plane whose intersection graph is connected. Compute the smallest cardinality subset $C' \subseteq C$ that separates s from t. They gave a polynomial time algorithm to solve this problem with a running time of $O(n^3)$. They also showed that the general point-separation problem is NP-hard for unit disks, and horizontal and vertical segments. Note that Penninger and Vigan [36] independently proved NP-hardness of the general point-separation problem for unit disks.

There has been some work in studying non-disk sensor networks. Tseng and Kirkpatrick [42] showed that determining resilience of a network of unit-line segments is APX-hard. They extended their NP-hardness proof to unit square sensors, and elliptical sensors and argued that their technique can be extended to any non-symmetrical shape sensors. Tseng [41] considered the question of given a sensor network of spheres in 3D space and two points s and t, what is the resilience of the senor network. They showed that computing this is NP-hard by a reduction from the vertex cover problem.

For further background refer to Wang's survey on coverage problems in sensor networks [45] (in particular, Section 6).

2.2 Minimum Shrinkage

We propose a new measure of barrier coverage, called *minimum shrinkage*. This new measure models the reality that a sensor's detection ability drops off with distance.

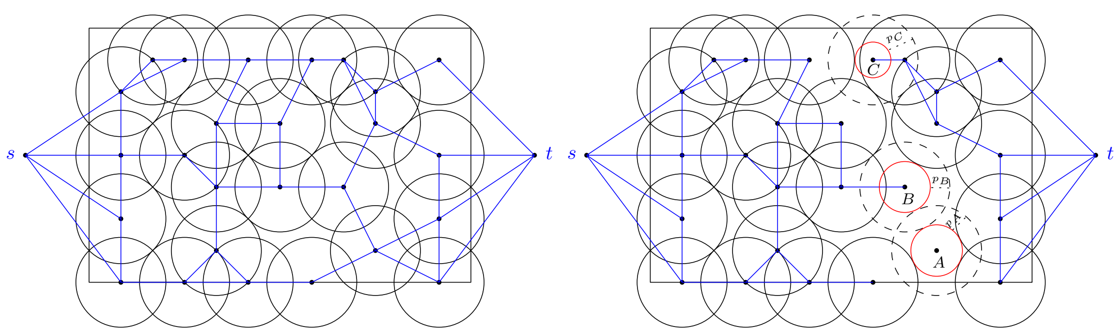
To *shrink* a unit disk by amount s_i for $0 \leq s_i \leq 1$ means to decrease its radius by s_i, i.e., to replace the unit disk by a disk of radius $(1 - s_i)$. Let c_i, $i = 1, \ldots, n$ be the locations of unit disk sensors in a sensor network that is supposed to act as a barrier between region R_1 and region R_2 in the plane. 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, $1 \leq i \leq n$, then the network no longer provides a barrier between regions R_1 and R_2, i.e., there is a path from R_1 to R_2 in the free space between the shrunken disks. See Figure 1.4.

2.2.1 Reduction from Minimum Shrinkage to *MSPEC*

We will show that the Minimum Shrinkage problem for rectangular barrier coverage can be formulated as an *MSPEC* problem.

Given a set of n points P in $\mathbb{R}^2$, the *unit disk graph* induced by P, denoted $UDG(P)$, is an embedded graph with vertex set P and an edge (u, v) when the unit disks centered at u and v intersect, i.e., the edge set is $\{(u, v) : u, v \in P \text{ and } \operatorname{dist}(u, v) \leq 2\}$.

An instance of the Minimum Shrinkage problem for rectangular barrier coverage consists of a set of points P inside a rectangle R. To reduce to *MSPEC* we start with the graph $UDG(P)$. Define the weight of edge (u, v) to be $(2 - \operatorname{dist}(u, v))$. This is the amount of shrinkage at u and v that is needed to make the disks at u and v non-intersecting. We add two special vertices s and t. Vertex s is connected to every vertex whose unit disk intersects the left boundary of R, and vertex t is connected to every vertex whose unit disk intersects the right boundary of R. To define the weight of these edges, let R_l and R_r be the lines through the left and right boundaries of R, respectively, and, for line L, let $\operatorname{dist}(u, L)$ be the distance from point u to line L. Define the weight of an edge (s, u) to be $(1 - \operatorname{dist}(u, R_l))$. This is the amount of shrinkage needed to make the disk at u not intersect the left boundary of R. Similarly, define the weight of an edge (t, u) to be $(1 - \operatorname{dist}(u, R_r))$. A solution to *MSPEC* on the resulting graph solves the Minimum Shrinkage problem where we interpret p_v as the amount to shrink the disk centered at v. We started this work originally by looking at the minimum shrinkage problem but due to this reduction, we know that the minimum shrinkage problem is a special case of *MSPEC*.



(a) (b)

Figure 2.2: Reduction from MSPEC to Minimum Shrinkage problem: (a) shows the intersection graph of unit disks. Weight of an edge (u, v) is $(2 - \operatorname{dist}(u, v))$. Vertex s and t are connected to vertices whose unit disk intersects the left and the right boundary of rectangle respectively; (b) shows shrunken disc for Minimum Shrinkage problem and the corresponding cut obtained for the MSPEC problem.

Observe that this reduction still works (with obvious modifications) for the more general problem where each sensor disk has a specified radius, i.e., the radii are not uniform.

Chapter 3

Approximation Scheme for *MSPEC*

In this chapter, we describe our approximation scheme for *MSPEC*. We give the precise construction of the algorithm. We also prove the relevant theorems needed to show that indeed our approximation algorithm is a $(1+\epsilon)$-approximation to *MSPEC*. We show that our algorithm is an FPTAS by carefully analyzing the running time of the algorithm.

The idea of our approximation algorithm is to convert the *MSPEC* problem to a minimum vertex cut problem. Observe that if we can only assign power 0 or 1 to every vertex in V, then our problem is Minimum Vertex Cut—remove a minimum number of vertices to disconnect s and t. We will discretize our problem 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. A similar approach was used, in a geometric setting, by Agarwal et al. [1]. We want to ensure that the discretization introduces an error of at most $\frac{\varepsilon}{n}$ for each vertex with respect to the optimum solution.

In order to carry out this plan, we need an upper bound on the maximum power we might assign to any vertex, and, for the error analysis, we need a lower bound on the optimum solution.

3.1 Minimum Bottleneck Shared-Power Edge Cut

To approximately solve the minimum shared-power edge cut problem, we will need a solution to another closely related problem which we refer to as the "Minimum Bottleneck Shared-Power Edge Cut Problem". This problem is very similar to *MSPEC* in that we want to assign powers to the vertices such that s and t become disconnected if we remove any edge where the sum of the powers on its endpoints is at least its weight. The key difference is that instead of assigning different powers to every vertex we will assign the same power to each vertex in V and minimize this "bottleneck" power. More precisely, we want the minimum value p such that s and t become disconnected if we remove the edges

$(u, v), u, v \in V$ with $2p \geq w_{u,v}$ and the edges $(u, v), u \in V, v \in \{s, t\}$ with $p \geq w_{u,v}$ (See Figure 3.1).

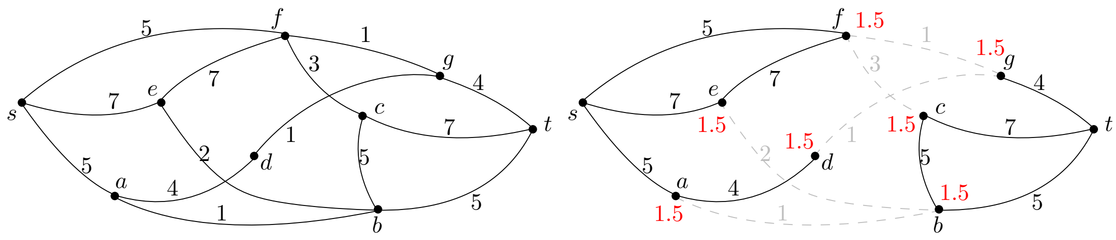


Figure 3.1: (a) Input graph G (b) Bottleneck power is shown in red. The cut edges are shown in light gray.

In the case of barrier coverage for a rectangular barrier, the minimum bottleneck shared-power edge cut is equivalent to the "maximum breach path" measure introduced by Meguerdichian et al. [29]. Namely, we want to shrink all sensor disks by the same (minimum) amount to permit a path in the free space between disks. Meguerdichian et al. computed the maximum breach path in polynomial time using Voronoi diagrams (See Figure 3.2). It is interesting that the problem can be solved in a more general non-geometric setting.

In the following we show how to obtain an exact solution to the minimum bottleneck shared-power edge cut in polynomial time. Let the set of removed edges in the optimal solution be M^* and the optimal power be p^*.

Claim 1. *There is an edge $e = (u, v) \in M^*$ which is* tight *for p^*, i.e., $w_{u,v} = 2p^*$ if $u, v \in V$ or $w_{u,v} = p^*$ if $u \in V$ and $v \in \{s, t\}$.*

Proof. Suppose $w_{u,v} < 2p^*$ for all edges in M^* with both end points in V and $w_{u,v} < p^*$ for all edges in M^* with one end point in V . Then we can reduce p^* to $(p^* - \varepsilon)$ and still ensure that $w_{u,v} \leq 2(p^* - \varepsilon)$ for all edges in M^* with both end points in V and $w_{u,v} \leq p^* - \varepsilon$ for edges in M^* with one end point in V, so p^* was not the minimum bottleneck power. $\square$

Theorem 6. *Minimum Bottleneck Shared-Power Edge Cut can be solved optimally in $O((n + m) \log m)$ time.*

Proof. Define the *power requirement* of edge $e = (u, v)$ to be $\frac{1}{2}w_{u,v}$ if $u, v \in V$ or $w_{u,v}$ if $u \in V$ and $v \in \{s, t\}$. By Claim 1, p^* is equal to the power requirement for some edge $e \in E$, and we can use binary search on the power requirements to find the minimum p such that s and t become disconnected if we remove the edges whose power requirement is at most p. We sort all the power requirements in non-decreasing order and perform binary search on this list. To test a value p' from this list, we create a graph G' with vertices of G and edges of G whose power requirement is greater than p'.

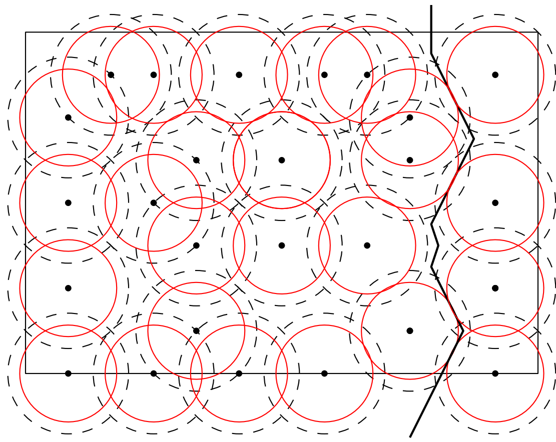


Figure 3.2: Maximum breach path travel on the Voronoi diagram of the sensor points. The shrunken discs are shown in red

If s and t are disconnected in G' we recursively perform a binary search on the power requirements between 0 and p', otherwise we recurse on all higher power requirements. The running time for this approach is determined by the binary search, which takes time $O(\log m)$, and by checking connectivity of s and t in G', which takes time $O(n+m)$. Therefore, the total running time is $O((n+m)\log m)$. Here n is the number of vertices and m is the number of edges. □

3.2 Approximation Scheme

Let $n = |V|$, let p^* be the minimum power for the bottleneck shared-power edge cut problem, and let OPT be the minimum power sum for $MSPEC$.

Lemma 7. *$p^* \leq \mathrm{OPT} \leq np^*$.*

Proof. Assigning power p^* to every vertex in V provides a feasible solution to $MSPEC$, and therefore $OPT \leq np^*$.

For the other inequality, let $p_{\max}$ be the maximum power assigned to any vertex of V in an optimum solution to $MSPEC$. Then $p_{\max} \leq OPT$. Assigning $p_{\max}$ to every vertex in V provides a solution to the minimum bottleneck shared-power edge cut. Therefore, $p^* \leq p_{\max} \leq OPT$. □

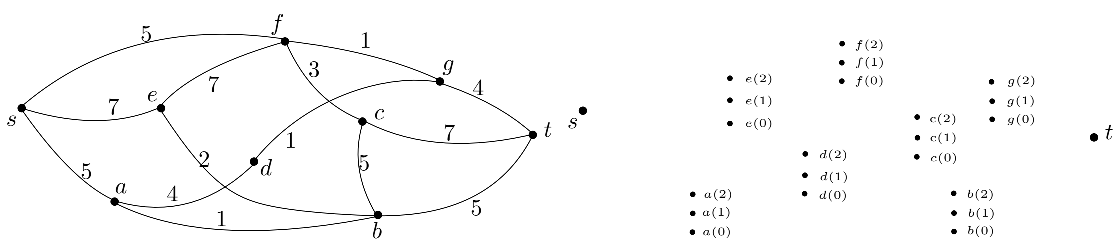
From this lemma, we know that the maximum power we might assign to a vertex is np^*. The lemma also implies that if we introduce an error of at most $\alpha = \frac{\varepsilon}{n}p^*$ for each vertex, then the total error over all vertices will be at most $\varepsilon p^* \leq \varepsilon OPT$. Our plan is to construct a new graph $G' = (V', E')$ in which we replace each vertex of V by c copies, where each copy represents power α. Since the total power that we might assign to the vertex is np^*, the number of copies of the vertex that we need is $c = \lceil \frac{np^*}{\alpha} \rceil = \lceil \frac{n^2}{\varepsilon} \rceil$. We will replace each vertex of V by a sequence of c vertices, $v(0), v(1), \ldots, v(c-1)$.

Removing the first k vertices of the sequence will correspond to assigning power $k\alpha$ to v. In G' we will assign edges to the copies $v(i)$ to reflect this. In particular, for $u, v \in V$ we put an edge $(u(i), v(j))$ in G' if $i\alpha + j\alpha < w_{u,v}$. For $u \in V, x \in \{s, t\}$ we put an edge $(u(i), x)$ in G' if $i\alpha < w_{u,x}$. A related idea of discretizing the choices using vertices combined with minimum cuts was developed by Hochbaum et al. [16, 17] for integer linear programs with at most two variables per inequality. They rely on an algorithm by Picard [37] for finding the minimum-cost closure of a directed graph.

The precise construction of G' is given in Algorithm 1.

Algorithm 1 Construction of G'

$\forall v \in V$, make $c = \lceil \frac{n^2}{\varepsilon} \rceil$ copies of v numbered $v(0), v(1), \ldots, v(c-1)$
$V' = \{v(i) \mid \forall v \in V, 0 \leq i < c\} \cup \{s, t\}$
$\alpha = \frac{\varepsilon p^*}{n}$
$E' = \{(u(i), v(j)) \mid i\alpha + j\alpha < w_{u,v}\} \cup \{(u(i), x) \mid x \in \{s, t\}, i\alpha < w_{u,x}\}$



(a) (b)

Figure 3.3: (a) Input graph G (b) Replace each vertex with $\lceil \frac{n^2}{\varepsilon} \rceil$ copies of it.

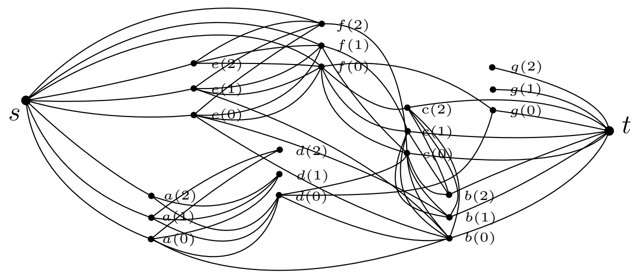


Figure 3.4: Add edges between $u(i)$ and $v(j)$ if $i\alpha + j\alpha < w_{u,v}$. Modified graph G' with $\alpha = 1$.

Our approximation algorithm now proceeds as follows. We find a minimum s-t vertex cut in G', denoted K^*, and use it to define power values on the vertices of G as given in Algorithm 2.

Algorithm 2 Approximation Algorithm for *MSPEC*

Construct G' as in Algorithm 1
$K^* \leftarrow$ minimum s-t vertex cut for G'
$k_v^* = |\{v(j) \in K^*\}|$ for $v \in V$
$p_v = k_v^* \cdot \alpha$ for $v \in V$
return $p = (p_v)_{v \in V}$ as the solution for *MSPEC* on G

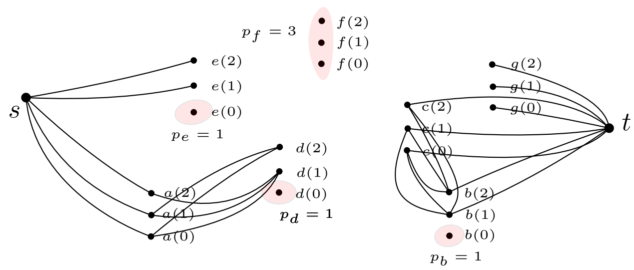


Figure 3.5: Approximate value of *MSPEC* can be found using the min s-t vertex cut of graph G'. $f(0), f(1), f(2), d(0), e(0)$, and $b(0)$ are in min s-t vertex cut.

To prove that this algorithm is an FPTAS we will first prove that the solution returned by the algorithm is indeed a feasible solution for *MSPEC*. Then, to bound the approximation factor, we will derive an upper bound on the solution returned by the algorithm in terms of the optimum solution to *MSPEC*. We first prove some properties of G' and of K^*.

Claim 2. *If $v \in V$ and $i_1 < i_2$, then in G' the neighbourhoods of $v(i_1)$ and $v(i_2)$ are related by $N(v(i_1)) \supseteq N(v(i_2))$.*

Proof. We will show that $(v(i_2), x) \in E'$ implies $(v(i_1), x) \in E'$.

Case 1. $x = u(j)$ for $u \in V$. Since the edge $(v(i_2), u(j))$ is in E' we have $i_2\alpha + j\alpha < w_{v,u}$ so $i_1\alpha + j\alpha < w_{v,u}$ which implies $(v(i_1), u(j)) \in E'$.

Case 2. $x \in \{s, t\}$. Since the edge $(v(i_2), x)$ is in E' we have $i_2\alpha < w_{v,x}$ so $i_1\alpha < w_{v,x}$ which implies $(v(i_1), x) \in E'$. □

Claim 3. *The copies of v in K^* are $v(0), v(1), \ldots, v(k_v^* - 1)$.*

Proof. We will prove that the copies of v in K^* form a prefix of $v(0), v(1), \ldots v(c-1)$. Then the result follows since there are k_v^* copies of v in K^*.

Consider $i_1 < i_2$. By Claim 2, $N(v(i_1)) \supseteq N(v(i_2))$. Now observe that if a graph contains vertices u and v with $N(u) \supseteq N(v)$ and v is in a minimum vertex cut, then u must be as well, since u is a duplicate of v with possibly some more edges. Therefore if $v(i_2)$ is in K^* then so is $v(i_1)$. □

Lemma 8. *A solution of Algorithm 2 is a feasible solution for MSPEC and the sum of the assigned powers is $\alpha|K^*|$.*

Proof. Let G_p be the result of removing from G all the edges (u, v) with $p_u + p_v \geq w_{u,v}$ where p is as defined by Algorithm 2. (Recall that we always set $p_s = p_t = 0$.) We must show that s and t are disconnected in G_p. Suppose not. Then there is an s-t path P in G_p, say $s, x_1, x_2, \ldots, x_l, t$. We will show that the path $P' = s, x_1(k^*_{x_1}), x_2(k^*_{x_2}), \ldots, x_l(k^*_{x_l}), t$ exists in $G' - K^*$, a contradiction to K^* being an s-t vertex cut.

First, note that all the vertices of P' lie in $G' - K^*$ by Claim 3. Consider an edge (u, v) of the path P with $u, v \in V$. Since this edge is still present in G_p, we know that $p_u + p_v < w_{u,v}$. Thus by the definition of the p values, $k^*_u\alpha + k^*_v\alpha < w_{u,v}$. By the construction of G', the edge $(u(k^*_u), v(k^*_v))$ is in E'.

It remains to consider the edges of P incident to s and t. Since the edge (s, x_1) is in G_p, we know that $p_{x_1} < w_{s,x_1}$. Thus by the definition of the p values, $k^*_{x_1}\alpha < w(s, x_1)$. By the construction of G', the edge $(s, x_1(k^*_{x_1}))$ is in E'. The argument for edge (x_l, t) is similar.

Thus the path P' exists in $G' - K^*$ which gives the desired contradiction.

In our solution we assign a power of p_v to vertex v, so the total power assigned is:

$$\sum_{v \in V} p_v = \sum_{v \in V} k^*_v \cdot \alpha = \alpha \sum_{v \in V} k^*_v = \alpha |K^*| \tag{3.1}$$

□

Next we prove an upper bound on the size of the set K^* (a minimum s-t vertex cut in G') in terms of an optimum solution to *MSPEC*. Let $p^*_v, v \in V$ be an optimum power assignment for *MSPEC*. Let $p^*_s = p^*_t = 0$.

Lemma 9. *G' has a vertex cut K of size $\sum \lceil \frac{p^*_v}{\alpha} \rceil$. Thus $|K^*| \leq \sum \lceil \frac{p^*_v}{\alpha} \rceil$.*

Proof. Let M^* be the set of edges of G that are removed by the optimum power assignment $p^*_v, v \in V$, i.e., $M^* = \{(u, v) : p^*_u + p^*_v \geq w_{u,v}\}$. M^* is an s-t cut in G.

Define $k_v = \lceil \frac{p^*_v}{\alpha} \rceil$ for $v \in V$. Define K, a set of vertices of G', to consist of the first k_v copies of each vertex $v \in V$, i.e., $K = \{v(i) : v \in V, 0 \leq i < k_v\}$. Observe that $|K| = \sum \lceil \frac{p^*_v}{\alpha} \rceil$. To prove the lemma, we just need to show that K is an s-t vertex cut in G'.

It suffices to show that if $(u, v) \in M^*$ then there is no copy of edge (u, v) in $G' - K$. Consider an edge $(u, v) \in M^*$. Then $p^*_u + p^*_v \geq w_{u,v}$, so $\lceil \frac{p^*_u}{\alpha} \rceil \alpha + \lceil \frac{p^*_v}{\alpha} \rceil \alpha \geq w_{u,v}$, i.e., $k_u\alpha + k_v\alpha \geq w_{u,v}$.

Now observe that if $i \geq k_u$ and $j \geq k_v$ then $i\alpha + j\alpha \geq k_u\alpha + k_v\alpha \geq w_{u,v}$, so $(u(i), v(j))$ is not an edge of G' (by definition of G'). On the other hand, if $i < k_u$ then $u(i) \in K$, so $u(i)$ is not a vertex of $G' - K$, and similarly, if $j < k_v$ then $v(j) \in K$, so $v(j)$ is not a vertex of $G' - K$. Thus no copy of the edge (u, v) exists in $G' - K$. This proves that K is an s-t vertex cut in G' and completes the proof of the lemma. □

Theorem 10. *Algorithm 2 is a fully polynomial time approximation scheme (FPTAS) for the Minimum Shared-Power Edge Cut problem (MSPEC). Furthermore, the running time of Algorithm 2 is $O(n^{5.5}m\varepsilon^{-2.5})$.*

Proof. By Lemma 8 we know that Algorithm 2 gives a solution to *MSPEC* of cost $\alpha|K^*|$, and by Lemma 9 we have $|K^*| \leq \sum \lceil \frac{p_v^*}{\alpha} \rceil$.

$$\begin{aligned}
\text{Value of approximate solution} &= \alpha|K^*| \leq \sum_{v \in V} \left\lceil \frac{p_v^*}{\alpha} \right\rceil \cdot \alpha \;<\; \sum_{v \in V} \left(\frac{p_v^*}{\alpha} + 1 \right) \cdot \alpha \\
&= \sum_{v \in V} \left(p_v^* + \alpha \right) \;=\; \sum_{v \in V} p_v^* + n\alpha \;=\; OPT + n\alpha.
\end{aligned}$$

Thus the approximation ratio is at most $\frac{OPT+n\alpha}{OPT} = \frac{OPT+\frac{n\varepsilon p^*}{n}}{OPT} = 1+\frac{\varepsilon p^*}{OPT} \leq (1+\varepsilon)$, since $OPT \geq p^*$. The running time of Algorithm 2 depends on the running time of finding a minimum s-t vertex cut and this can be done in time $O(n^{1/2}m) = O(n^{2.5})$ for a graph with n vertices and m edges using a modified version of Dinitz's flow algorithm [31, Chapter 1] [39, Corollary 9.7a]. If our original graph G has n vertices and m edges, then the graph G' has $|V'| = nc = O(\frac{n^3}{\varepsilon})$ vertices and at most $|E'| \leq |E|c^2 = O(\frac{mn^4}{\varepsilon^2})$ edges, and Algorithm 2 finds a vertex cut in graph G' in $O(n^{5.5}m\varepsilon^{-2.5})$ time. $\square$

Chapter 4

Variations of *MSPEC*

In this chapter we give polynomial time algorithms for three special cases of *MSPEC*: *MSPEC* with uniform edge weights, *MSPEC* with integral edge weights, and when the power values at each vertex are restricted to a polynomially bounded domain. In Section 4.3 we show that our FPTAS extends to a general version of *MSPEC* where there are costs on vertices.

4.1 Uniform/Integral edge weights

We use the alternative formulation of *MSPEC* given at the start of Section 1.3. For any instance of *MSPEC* there is a vertex partition S, T with $s \in S$ and $t \in T$, such that the minimum power sum is equal to the minimum "w-vertex cover" in the bipartite graph B of edges between S and T, and this is equal to the weight of a maximum matching in B. By Kőnig's Theorem (see Theorem 2) and its generalization to weights (Egerváry's Theorem, see Theorem 3) we know that if the edge weights are integral then there is an optimal solution where the power values are also integral, and if the edge weights are all 1 then it suffices to consider power values that are $\{0, 1\}$-valued.

For *MSPEC* with uniform edge weights, we can scale so that all edge weights are 1. Then, by the above, the power values will be 0 or 1 and the problem reduces to Minimum Vertex Cut, which can be solved in time $O(n^{1/2}m)$ time as discussed in the previous section.

For *MSPEC* with integer edge weights, we can assume that the power values are integral. Furthermore, if the edge weights are bounded by W, then so are the power values. We can then use the same approach as Algorithm 1, but make $W + 1$ copies of each vertex, and set $\alpha = 1$. This solves *MSPEC* exactly, with a running time of $O((nW)^{1/2}(mW^2)) = O(n^{1/2}mW^{5/2})$. Thus it provides a pseudo polynomial time algorithm for *MSPEC* with integral edge weights.

4.2 Polynomial Domain

In activation network design problems, one of the most common assumptions is that of a 'polynomial domain' for the power values (See Subsection 1.2.5 or [34, 32]). This means that the power values for vertex v come from a set D^v whose size is bounded by a polynomial in $n = |V|$. The polynomial domain assumption is realistic for wireless networks when there are a small number of possible powers that can be assigned to a vertex. However, this does not apply to the minimum shrinkage problem because we are using shrinkage as a measure of barrier coverage rather than making any assumption about sensor powers.

Under the polynomial domain assumption, *MSPEC* admits a polynomial time algorithm. We sketch the algorithm. The technique is similar to the one we used in the previous section. We assume $0 \in D^v$ for all v (otherwise, we must pay $\min(D^v)$ directly, and can then subtract it from all later values). Let the values in D^v be $d_0^v, d_1^v, \ldots, d_{c_v}^v$ in increasing order. We modify the construction of graph G' from Algorithm 1, to create $|D^v| = c_v + 1$ copies of each vertex v, which we number $v(0), v(1), \ldots, v(c_v)$. We assign a cost of $d_{i+1}^v - d_i^v$ to $v(i)$ for $i = 0, \ldots, c_v - 1$, and ∞ to $v(c_v)$.

In graph G' we add an edge between two vertices $u(i)$ and $v(j)$ if $d_i^u + d_j^v < w_{u,v}$. We now run a max flow algorithm to find the minimum cost s-t vertex cut in G' by applying the standard graph transformation (see [46]) to convert a vertex cut to an edge cut, and vertex weights to edge capacities. If the minimum cost is ∞ then there is no feasible assignment of powers from the given domains. Otherwise, for vertex v, if i is the maximum index for which $v(i)$ is included in the cut, then we assign $p_v = d_{i+1}^v$. Note that a feasible solution has $i < c_v$ so this is well-defined.

For correctness, first observe that the prefix property, as mentioned in Claim 3, still holds. Next, we claim that if the assigned powers do not remove edge (u, v) then there was a copy of that edge in G' after removing the optimum vertex cut. Suppose we assigned $p_u = d_{i+1}^u$ and $p_v = d_{j+1}^v$ but $p_u + p_v < w_{u,v}$. Then the edge $(u(i+1), v(j+1))$ still exists in G'. In the other direction, a solution to *MSPEC* with powers from the given domains yields a vertex cut in G' of equal cost.

The running time for the algorithm is determined by the running time of the standard max-flow algorithms, which is $O(n^3/\log n)$ [10] (or see [39, Section 10.8]). The number of vertices in the new graph G' is $\sum_v |D^v|$, thus the running time is $O((\sum_v |D^v|)^3/\log(\sum_v |D^v|))$.

4.3 *MSPEC* with Vertex Costs

While trying to prove NP-hardness for *MSPEC*, we thought it might be easier to prove the hardness result for a more general version of *MSPEC* where each vertex also has a multiplicative cost associated with it. Though we were unsuccessful in proving the hardness, we were able to extend our FPTAS result to even this case. Formally, *MSPEC* with multiplicative vertex costs problem is defined as:

Input: Graph $G = (V \cup \{s,t\}, E)$ with a non-negative weight $w_{u,v}$ on each edge $(u,v) \in E$ and a non-negative cost c_v on every vertex $v \in V$.

Problem: Assign a non-negative power p_v to each $v \in V$ and assign $p_s = p_t = 0$ such that removing the edge set $\{(u,v) \in E : p_u + p_v \geq w_{u,v}\}$ disconnects s and t and $\sum_{v \in V} c_v p_v$ is minimized.

Analogously to Section 3.1, we define a bottleneck version of *MSPEC* with multiplicative vertex costs, where we want to pay the same cost $k = p_v c_v$ at each vertex $v \in V$. For the same input as above, we want the minimum value k such that s and t become disconnected if we remove the edges $(u,v), u, v \in V$ with $\frac{k}{c_u} + \frac{k}{c_v} \geq w_{u,v}$ and the edges $(u,v), u \in V, v \in \{s,t\}$ with $\frac{k}{c_u} \geq w_{u,v}$.

We show how to obtain an exact solution to Minimum Bottleneck Shared-Power Edge Cut with vertex costs in polynomial time. Let the set of removed edges in the optimal solution for Minimum Bottleneck Shared-Power Edge Cut with vertex costs be M^* and the optimal cost be k^*.

Claim 4. *There is an edge $e = (u,v) \in M^*$ which is* tight *for k^*, i.e., $w_{u,v} = \frac{k^*}{c_u} + \frac{k^*}{c_v}$ or $w_{u,v} = \frac{k^*}{c_u}$ if $u \in V$ and $v \in \{s,t\}$.*

Proof. Suppose $w_{u,v} < \frac{k^*}{c_u} + \frac{k^*}{c_v}$ for all edges in M^* with both end points in V and $w_{u,v} < \frac{k^*}{c_u}$ for all edges in M^* with one end point in V. Then we can reduce k^* to $(k^* - \varepsilon)$ and still ensure that $w_{u,v} \leq \frac{k^*-\varepsilon}{c_u} + \frac{k^*-\varepsilon}{c_v}$ for all edges in M^* with both end points in V and $w_{u,v} \leq \frac{k^*-\varepsilon}{c_u}$ for edges in M^* with one end point in V, so c^* was not the minimum bottleneck cost.

□

Theorem 11. *Minimum Bottleneck Shared-Power Edge Cut with vertex costs can be solved optimally with a running time of $O((n+m)\log m)$.*

Proof. Define the *cost requirement* of an edge $e = (u,v)$ to be $\frac{w_{u,v}}{\frac{1}{c_u} + \frac{1}{c_v}}$ if $u, v \in V$ or $w_{u,v} c_u$ if $u \in V$ and $v \in \{s,t\}$. By Claim 4, k^* is equal to the cost requirement for some edge $e \in E$, i.e., $w_{u,v} = \frac{k^*}{c_u} + \frac{k^*}{c_v}$ or $w_{u,v} = \frac{k^*}{c_u}$ which implies that $k^* = \frac{w_{u,v}}{\frac{1}{c_u} + \frac{1}{c_v}}$ if $u, v \in V$ or $k^* = w_{u,v} c_u$ if $u \in V$ and $v \in \{s,t\}$. We can use binary search on the cost requirements to find the minimum k such that s and t become disconnected if we remove the edges whose cost requirement is at most k.

Similar to the proof of Theorem 6, we do a binary search on the sorted (in non-decreasing order) list of cost requirements instead of the power requirements to find the optimal cost. This can be done in $O((n+m)\log m)$ running time. □

Let *OPT* be the optimal solution of *MSPEC* with vertex costs and let k^* be the optimal solution of bottleneck *MSPEC* with vertex costs.

Lemma 12. $k^* \leq \mathrm{OPT} \leq nk^*$.

Proof. Assigning cost k^* to every vertex in V provides a feasible solution to *MSPEC* with vertex costs, therefore, $OPT \leq nk^*$.

For the other inequality, let $k_{\max}$ be the maximum cost assigned to any vertex of V in OPT. Then $k_{\max} \leq OPT$. Assigning $k_{\max}$ to every vertex in V provides a solution to Minimum Bottleneck Shared-Power Edge Cut with vertex costs. Therefore, $k^* \leq k_{\max} \leq OPT$. □

To get an FPTAS for *MSPEC* with vertex costs we modify the construction of graph $G' = (V', E')$ in Algorithm 1. In V', we still create $\lceil \frac{n^2}{\varepsilon} \rceil$ copies of every vertex in V. We set α equal to $\frac{\varepsilon k^*}{n}$ and define the edge set $E' = \{(u(i), v(j)) \mid \frac{i\alpha}{c_u} + \frac{j\alpha}{c_v} < w_{u,v}\} \cup \{(u(i), x) \mid x \in \{s, t\}, \frac{i\alpha}{c_u} < w_{u,x}\}$. The remainder of the algorithm and the proofs carry over after similar modifications.

Chapter 5

Faster Approximation for *MSPEC*

In this chapter, we will discuss two different approaches to speed up the running time of our approximation algorithm. For the first approach, we show that a discrete version of *MSPEC* is a 2-approximate solution for *MSPEC*. Using this solution, we design a faster FPTAS for *MSPEC* with a running time of $O(n^3 m\varepsilon^{-2.5} + m^3/\log m)$. In the second approach, using max power s-t cut as a 2-approximate solution for *MSPEC* we manage to shave off $O(m^3/\log(m))$ from the running time of our FPTAS.

5.1 Speedup using the discrete version of *MSPEC*

In this section, we will show how to speed up our FPTAS from the current running time of $O(n^{5.5} m\varepsilon^{-2.5})$ to $O(n^3 m\varepsilon^{-2.5} + m^3/\log m)$. To accomplish this, we discretize the problem using a faster 2-approximation for *MSPEC* instead of the n-approximation given by the minimum bottleneck shared power edge cut. This kind of approach is well studied in the literature, for instance Benkoczi et al. [5] improved the running time of a $(1+\epsilon)$-approximation algorithm for "minimizing total sensor movement for barrier coverage by non-uniform sensors on a line" from $O(n^7\epsilon^{-3})$ [4] to $O(n^5\epsilon^{-3})$.

For the 2-approximation, we consider a discrete version of *MSPEC* where a vertex can only be assigned a power equal to the weight of one of its incident edges. With this restriction, the number of possible choices of power assignment at a given vertex is equal to the degree of the vertex in G.

Thus the polynomial domain assumption is satisfied and the algorithm of Section 4.2 solves this discrete version in time $O((\sum_v \deg(v))^3/\log(\sum_v \deg(v))) = O(m^3/\log m)$.

We now use the fact that the discrete *MSPEC* provides a 2-approximation. This was proved more generally by Panigrahi [34, Lemma 3.11], but we include his short argument. Let *OPT* be the optimum solution value for *MSPEC*.

Theorem 13 ([34])**.** *Discrete* MSPEC *is a 2-approximation for* MSPEC*.*

Proof. We will show that $2OPT$ is an upper bound for the discrete version of *MSPEC*, which proves the theorem. Let C be the set of edges in the s-t cut determined by an optimum solution. For each edge $e = (u, v) \in C$, select the endpoint v of higher power. Then $p_v \geq \frac{1}{2}w_{u,v}$. For each selected vertex we raise p_v to the maximum $w_{x,v}$ where (x, v) is an edge of C that selects v. Then we at most double the power of v. Furthermore, any vertex not selected can have its power decreased to 0. The new power values still activate all edges of C and provide a solution to discrete *MSPEC*. □

Let Z be the optimal value of a solution to the discrete *MSPEC* problem on graph G. Clearly $OPT \leq Z$, and from Theorem 13, we have $Z \leq \ 2OPT$. Thus:

$$\frac{Z}{2} \leq OPT \leq Z \tag{5.1}$$

We can use these bounds in place of those in Lemma 7 to obtain a faster approximation algorithm for *MSPEC*. From Equation (5.1), the maximum power we assign to any vertex is Z. If we introduce an error of at most $\alpha = \frac{\varepsilon Z}{2n}$ at each vertex, the total error will be at most $n\alpha \leq \varepsilon OPT$. We construct the graph G' as in Algorithm 1, though instead of taking $\lceil \frac{n^2}{\varepsilon} \rceil$ copies of each vertex, we only use $c' = \lceil \frac{Z}{\alpha} \rceil = \lceil \frac{2Zn}{\varepsilon Z} \rceil = \lceil \frac{2n}{\varepsilon} \rceil$ copies.

Algorithm 2 on this new G' is an FPTAS with a running time of $O(|V'|^{1/2}|E'|) = O((n\frac{n}{\varepsilon})^{1/2}(m(\frac{n}{\varepsilon})^2)) = O(n^3 m\varepsilon^{-2.5})$. Since the time to compute Z is $O(m^3/\log(m))$, the overall running time of our fast FPTAS is $O(n^3 m\varepsilon^{-2.5} + m^3/\log m)$. In addition to improving the running time, this algorithm also improves the space complexity by at least $O(n^2)$.

5.2 Speedup using Min-Power Cover Cut

We use ideas similar to those in the previous section to get a faster FPTAS. In particular, we shave off $O(m^3/\log(m))$ from the previous running time to get an FPTAS with a running time of $O(n^3 m\varepsilon^{-2.5})$. The plan is to use a faster 2-approximation algorithm for *MSPEC* and then use a similar discretization approach as of Section 5.1.

The faster 2-approximation relies on an algorithm by Angel et al. [3] to find a minimum edge cut under a different power model. Recall from Subsection 1.2.3 the "Max Power Optimization" model (2) where an edge $e = (u, v)$ is said to be activated if $\max\{p_u, p_v\} \geq w_{u,v}$. In this model the edge cut problem becomes: Given a graph with distinguished vertices s and t, assign powers to the vertices so that the "activated" edges with $\max\{p_u, p_v\} \geq w_{u,v}$ form an s-t cut, and the sum of powers is minimized. We will call this the *MMPEC* problem, for "Min Max-Power Edge Cut". Angel et al. [3] called it the "Min-Power-Cover cut problem", and gave an optimal algorithm with running time $O(m^{1/2}m) = O(m^{3/2})$.

We will now show that an optimal solution to *MMPEC* is a 2-approximation to *MSPEC*. We begin with a claim that any edge cut provides a feasible solution to *MMPEC*.

Algorithm 3 Greedy power assignment for a given cut C

Let C be an s-t cut.
Initialize set S to consist of the edges of C initialize M' to $\{\}$.
while S is not empty **do**
  Choose the edge $(u, v) \in S$ with max weight $w_{u,v}$ and assign $p_u = p_v = w_{u,v}$.
  Update $M' \leftarrow M' \cup (u, v)$.
  Delete all the edges in S incident to u and v including (u, v).
end while

Claim 5. *The greedy power assignment (Algorithm 3) is a feasible solution for MMPEC. The greedy power assignment computes powers and a matching M' that provide a feasible solution to MMPEC with objective value $2 \times \sum_{e \in M'} w_e$.*

Proof. Notice that if we can remove all the edges of C by assigning sufficient power to the vertices in V, then we are done. Let us consider the i^{th} iteration of the greedy algorithm where we picked the max weight edge $(u, v) \in S$, assigned power $w_{u,v}$ to both u and v, and deleted all the edges in S incident to either u or v. The weight of all edges that were deleted in this iteration was less than or equal to $w_{u,v}$. Since all the deleted edges had at least one of their endpoints as either u or v, assigning $w_{u,v}$ to u and v is sufficient to delete them under the max power optimization model. Given an s-t cut C, by the greedy assignment of powers, we get a feasible solution for *MMPEC*.

Notice that M' is a matching of cut edges. In the i^{th} iteration, we take the edge (u, v) and delete all the edges incident to either u or v. All edges that are selected in subsequent iterations have no endpoint common either with u or v.

By the greedy assignment, we assign a total power of $\sum_{(u,v) \in M'} 2w_{u,v}$. □

Theorem 14. *A solution to MMPEC is a 2-approximation for MSPEC.*

Proof. Consider the alternative formulation of *MSPEC* in terms of the matching (as defined in Subsection 1.3.1), i.e., partition the vertices of G into two sets with s in one set and t in the other, and minimize the weight of a maximum matching of the edges crossing between the two sets. Let the edges in an optimal cut for *MSPEC* be C and the max weight matching of C be M. Then $OPT_{MSPEC} = \sum_{e \in M} w_e$.

Consider the greedy matching M' of C as computed in Claim 5. By Claim 5, there is a feasible solution of *MMPEC* with a cost of $\sum_{e \in M'} 2w_e$. Therefore, $OPT_{MMPEC} \le \sum_{e \in M'} 2w_e = 2 \times \sum_{e \in M'} w_e$.

M is a max-weight matching of C, therefore, $\sum_{e \in M'} w_e \le \sum_{e \in M} w_e$. We get following chain of inequalities:

$$OPT_{MMPEC} \le 2 \times \sum_{e \in M'} w_e \le 2 \times \sum_{e \in M} w_e = 2 \times OPT_{MSPEC} \tag{5.2}$$

□

Let Z be the optimal solution of $MMPEC$ and OPT be the optimal solution of $MSPEC$. From Theorem 14, we have $Z \leq 2OPT$. Since a feasible solution for $MMPEC$ is also a feasible solution for $MSPEC$, we get $OPT \leq Z$ and combining the above two gives:

$$\frac{Z}{2} \leq OPT \leq Z \tag{5.3}$$

This approach can be used to compute a $(1 + \epsilon)$-solution as done in Section 5.1. A 2-approximate solution using $MMPEC$ can be found with a running time of $O(m^{3/2})$. This gives us an FPTAS for $MSPEC$ with a running time of $O(n^3 m\epsilon^{-2.5})$.

Chapter 6

Conclusion and Open Problems

We introduced a new problem called the Minimum Shared-Power Edge Cut (*MSPEC*). The goal is to assign powers at the vertices and remove an edge if the sum of powers on vertices is at least the weight of the edge between them. The original motivation for this problem comes from studying Minimum Shrinkage problem. Minimum Shrinkage problem is a more practical measure of classic barrier coverage problem.

We also studied several other variants of *MSPEC* that include uniform edge weights, *MSPEC* under the polynomial domain assumption, and vertex weighted *MSPEC*. For *MSPEC*, *minimum shrinkage* problem, and vertex weighted *MSPEC* we gave a fully polynomial time approximation scheme. For uniform edge-weighted *MSPEC*, we showed that the problem can be solved in pseudo-polynomial time. Under the polynomial domain assumption, we showed that *MSPEC* admits a polynomial time solution.

We hope that this work will help to unify barrier coverage problems with activation network design problems and foster a new research direction.

The biggest open question is to settle the complexity of the *MSPEC* problem—is it in P? NP-hard? For hardness, a starting point would be to show that *MSPEC* with vertex costs is NP-hard. On the other side, the complexity of the minimum shrinkage problem (i.e., the special case of *MSPEC* arising from shrinking unit disks in the plane) is also open. Can the geometry of unit disk graphs be used to develop a polynomial time algorithm?

Bibliography

[1] Pankaj K. Agarwal, Sariel Har-Peled, Haim Kaplan, and Micha Sharir. Union of random Minkowski sums and network vulnerability analysis. *Discrete & Computational Geometry*, 52(3):551–582, 2014.

[2] Ernst Althaus, Gruia Calinescu, Ion I. Mandoiu, Sushil Prasad, Nickolay Tchervenski, and Alexander Zelikovsky. Power efficient range assignment for symmetric connectivity in static ad hoc wireless networks. *Wireless Networks*, 12(3):287–299, 2006.

[3] Eric Angel, Evripidis Bampis, Vincent Chau, and Alexander Kononov. Min-power covering problems. In *Proceedings of the International Symposium on Algorithms and Computation (ISAAC)*, pages 367–377. Springer, 2015.

[4] Robert Benkoczi, Zachary Friggstad, Daya Gaur, and Mark Thom. Minimizing total sensor movement for barrier coverage by non-uniform sensors on a line. In *Revised Selected Papers of the 11th International Symposium on Algorithms for Sensor Systems (ALGOSENSORS) - Volume 9536*, pages 98–111. Springer-Verlag, 2015.

[5] Robert Benkoczi, Daya Ram Gaur, and Mark Thom. A 2-approximation algorithm for barrier coverage by weighted non-uniform sensors on a line. In *Algorithms for Sensor Systems - 12th International Symposium on Algorithms and Experiments for Wireless Sensor Networks (ALGOSENSORS)*, pages 95–111. Springer-Verlag, 2016.

[6] Sergey Bereg and David Kirkpatrick. Approximating barrier resilience in wireless sensor networks. In Shlomi Dolev, editor, *Algorithmic Aspects of Wireless Sensor Networks*, pages 29–40. Springer-Verlag, 2009.

[7] Sergio Cabello and Panos Giannopoulos. The complexity of separating points in the plane. *Algorithmica*, 74(2):643–663, 2016.

[8] Sergio Cabello, Kshitij Jain, Anna Lubiw, and Debajyoti Mondal. Minimum shared-power edge cut. *CoRR*, abs/1806.04742, 2018.

[9] Gruia Calinescu and Peng-Jun Wan. Range assignment for high connectivity in wireless ad hoc networks. In Samuel Pierre, Michel Barbeau, and Evangelos Kranakis, editors, *Ad-Hoc, Mobile, and Wireless Networks*, pages 235–246. Springer-Verlag, 2003.

[10] Joseph Cheriyan, Torben Hagerup, and Kurt Mehlhorn. An $o(n^3)$-time maximum-flow algorithm. *SIAM Journal on Computing*, 25(6):1144–1170, 1996.

[11] Hristo N. Djidjev. Approximation algorithms for computing minimum exposure paths in a sensor field. *ACM Transactions on Sensor Networks (TOSN)*, 7(3):23, 2010.

[12] Amitabha Ghosh and Sajal K. Das. Coverage and connectivity issues in wireless sensor networks: A survey. *Pervasive and Mobile Computing*, 4(3):303 – 334, 2008.

[13] Matt Gibson, Gaurav Kanade, Rainer Penninger, Kasturi Varadarajan, and Ivo Vigan. On isolating points using unit disks. *Journal of Computational Geometry*, 7(1):540–557, 2016.

[14] Matt Gibson, Gaurav Kanade, and Kasturi Varadarajan. On isolating points using disks. In *Proceedings of the 19th European Conference on Algorithms (ESA)*, pages 61–69. Springer-Verlag, 2011.

[15] Mohammad T. Hajiaghayi, Guy Kortsarz, Vahab S. Mirrokni, and Zeev Nutov. Power optimization for connectivity problems. *Mathematical Programming*, 110(1):195–208, 2007.

[16] Dorit S. Hochbaum, Nimrod Megiddo, Joseph Seffi Naor, and Arie Tamir. Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality. *Mathematical Programming*, 62(1-3):69–83, 1993.

[17] Dorit S. Hochbaum and Joseph Naor. Simple and fast algorithms for linear and integer programs with two variables per inequality. *SIAM J. Comput.*, 23(6):1179–1192, 1994.

[18] Alan J. Hoffmann and Joseph B. Kruskal. Integral boundary points of convex polyhedra. *Linear inequalities and related systems*, pages 223–246, 1956.

[19] Masao Iri. A new method of solving transportation-network problems. *Journal of the Operations Research Society of Japan*, 3(1):2, 1960.

[20] Philip Klein and R. Ravi. A nearly best-possible approximation algorithm for node-weighted steiner trees. *J. Algorithms*, 19(1):104–115, July 1995.

[21] Matias Korman, Maarten Löffler, Rodrigo I. Silveira, and Darren Strash. On the complexity of barrier resilience for fat regions and bounded ply. *Computational Geometry*, 72:34 – 51, 2018.

[22] Guy Kortsarz and Zeev Nutov. Approximating minimum-power edge-covers and 2,3-connectivity. *Discrete Appl. Math.*, 157(8):1840–1847, April 2009.

[23] Harold W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2):83–97, 1955.

[24] Harold W. Kuhn. On combinatorial properties of matrices. *George Washington University Logistics Papers*, 11:1–11, 1955.

[25] Santosh Kumar, Ten H. Lai, and Anish Arora. Barrier coverage with wireless sensors. *Wireless Networks*, 13(6):817–834, 2007.

[26] Yuval Lando and Zeev Nutov. On minimum power connectivity problems. *J. of Discrete Algorithms*, 8(2):164–173, June 2010.

[27] Benyuan Liu, Olivier Dousse, Jie Wang, and Anwar Saipulla. Strong barrier coverage of wireless sensor networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pages 411–420. ACM, 2008.

[28] Seapahn Megerian, Farinaz Koushanfar, Gang Qu, Giacomino Veltri, and Miodrag Potkonjak. Exposure in wireless sensor networks: Theory and practical solutions. *Wireless Networks*, 8(5):443–454, 2002.

[29] Seapahn Meguerdichian, Farinaz Koushanfar, Gang Qu, and Miodrag Potkonjak. Exposure in wireless ad-hoc sensor networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 139–150. ACM, 2001.

[30] Joseph S.B. Mitchell. On maximum flows in polyhedral domains. *Journal of Computer and System Sciences*, 40(1):88–123, 1990.

[31] Hiroshi Nagamochi and Toshihide Ibaraki. *Algorithmic Aspects of Graph Connectivity*. Cambridge University Press, 2008.

[32] Zeev Nutov. Survivable network activation problems. *Theoretical Computer Science*, 514:105–115, 2013.

[33] Zeev Nutov. Activation network design problems. In Teofilo F. Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics, Second Edition*, chapter 15. CRC Press, 2018.

[34] Debmalya Panigrahi. Survivable network design problems in wireless networks. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1014–1027, 2011.

[35] Debmalya Panigrahi. *Optimization Problems in Network Cconnectivity*. PhD thesis, Massachusetts Institute of Technology, 2012.

[36] Rainer Penninger and Ivo Vigan. Point set isolation using unit disks is NP-complete. *CoRR*, abs/1303.2779, 2013.

[37] Jean-Claude Picard. Maximal closure of a graph and applications to combinatorial problems. *Management Science*, 22(11):1268–1272, 1976.

[38] Rizzi Romeo. A short proof of König's matching theorem. *Journal of Graph Theory*, 33(3):138–139.

[39] Alexander Schrijver. *Combinatorial Optimization: Polyhedra and Efficiency*, volume A. Springer Science & Business Media, 2003.

[40] Gilbert Strang. Maximal flow through a domain. *Mathematical Programming*, 26(2):123–143, 1983.

[41] Kuan-Chieh Robert Tseng. *Resilience of Wireless Sensor Networks*. PhD thesis, University of British Columbia, 2011.

[42] Kuan-Chieh Robert Tseng and David Kirkpatrick. On barrier resilience of sensor networks. In Thomas Erlebach, Sotiris Nikoletseas, and Pekka Orponen, editors, *Algorithms for Sensor Systems*, pages 130–144. Springer Berlin Heidelberg, 2012.

[43] Vijay V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2013.

[44] Giacomino Veltri, Qingfeng Huang, Gang Qu, and Miodrag Potkonjak. Minimal and maximal exposure path algorithms for wireless embedded sensor networks. In *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 40–50. ACM, 2003.

[45] Bang Wang. Coverage problems in sensor networks: A survey. *ACM Computing Surveys*, 43(4):32:1–32:53, October 2011.

[46] Douglas B. West. *Introduction to Graph Theory*. Pearson, 2000.