Minimum shared-power edge cut

Sergio Cabello1 | Kshitij Jain2 | Anna Lubiw3 | Debajyoti Mondal4

1Department of Mathematics, University of Ljubljana and IMFM, Ljubljana, Slovenia
2Borealis AI, Waterloo, Canada
3School of Computer Science, University of Waterloo, Waterloo, Canada
4Department of Computer Science, University of Saskatchewan, Saskatoon, Canada

Correspondence
Anna Lubiw, University of Waterloo, Waterloo, Canada.
Email: alubiw@uwaterloo.ca

Funding information
This research was supported by the Javna Agencija za Raziskovalno Dejavnost RS, J1-813, J1-8155, P1-029. Natural Sciences and Engineering Research Council of Canada.

Abstract
We introduce a problem called 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, that is, without entering any disk. 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 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.

KEYWORDS
activation network, barrier coverage, fully polynomial-time approximation scheme, s-t cut, wireless network

1 INTRODUCTION

Minimum weight edge cuts in graphs are very well studied. In this paper we look at a variation that arises from unit disk graphs 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 \( e = uv \) 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:** Undirected graph \( G = (V \cup \{s, t\}, E) \) with \( n \) vertices, \( m \) edges, and a nonnegative weight \( w_{uv} \) on each edge \( uv \in E \).

**Problem:** Assign a nonnegative power \( p_v \) to each \( v \in V \) and assign \( p_s = p_t = 0 \) so that removing the edge set \( \{uv \in E : p_u + p_v \geq w_{uv}\} \) disconnects \( s \) and \( t \) and \( \sum_{v \in V} p_v \) is minimized.

Our main result is a fully polynomial time approximation scheme (FPTAS) for the MSPEC 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 [34]. 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 and commonly used measure of barrier coverage, introduced by Kumar et al. [20], is the minimum number of disks whose removal permits a path from \( R_1 \) to \( R_2 \) in the free space outside the disks. It is
not known if computing this measure is NP-hard or in P. However, there is a polynomial time algorithm [20] 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.

We suggest, as a more sophisticated measure of barrier coverage, the minimum total amount by which we must shrink the disks to permit a path from region $R_1$ to region $R_2$. We call this “minimum shrinkage”—it reflects the reality that sensor strength typically deteriorates (“attenuates”) with distance from the sensor [34]. Informally, minimum shrinkage measures the strength of the barrier by estimating the possibility that the “best” path through the sensors is detected, taking into account that a path that goes very close to a sensor has a greater chance of detection than a path that only skirts the edge of the unit disk. We use the sum of shrinkages to obtain a cumulative measure, as in previous work [20, 22]; more complicated measures that take exposure time into account are discussed in Section 3.1. We model sensor deterioration linearly. More realistically, one should use an inverse power with an exponent between 1 and 4 [22], but we leave this for future work.

Recently Cabello and Colin de Verdière [6] showed that the general minimum shrinkage problem is weakly NP-hard. For a rectangular barrier, the minimum shrinkage problem (as in Figure 1) 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 (see the details in Section 3.3). It is open whether minimum shrinkage for a rectangular barrier is in P or NP-complete. Our main result provides an approximation scheme to compute Minimum Shrinkage for rectangular barrier coverage.

1.1 Related models and work

Motivated by problems in the 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 must form a spanning tree, or a $k$-connected subgraph. The models that have been considered for selecting, or “activating,” an edge $uv$ of weight $w_{uv}$ are:

1. $\min\{p_u, p_v\} \geq w_{uv}$ ("power optimization", e.g., [13])
2. $\max\{p_u, p_v\} \geq w_{uv}$ ([4])
3. $p_u + p_v \geq w_{uv}$ ("installation cost" [28])
4. a more general function of $p_u$ and $p_v$ ("activation networks" [28])

Our model is (3). We briefly survey results on the above models. Of particular note is the fact that many of the positive results depend on a property/assumption that power values come from a discrete set—an assumption that we do not make because it is not justified in our geometric motivating problem. As discussed below, the only result for our problem of $s$-$t$ cuts was in model (2).

Model (1), “power optimization,” has been heavily studied. In this model, 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, which makes the problems easier. The literature on power optimization includes approximation algorithms for minimum spanning tree, Steiner forest, $k$-vertex or $k$-edge connected subgraph (all-pairs, single source, or $s$-$t$), and various degree-constrained and edge-cover problems [3, 10, 13, 21].

Model (2) was defined by Angel et al. [4]. As in the previous model, the 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, Angel et al. reduced the problem of finding an optimum $s$-$t$ cut in this model to the conventional minimum cut problem. This problem...
is closest to ours, but their model differs from ours, so their result does not apply in our setting. They also gave a polynomial
time algorithm for optimum \( s-t \) path, a 2-approximation for vertex cover, and a proof that spanning tree is hard to approximate.

Model (3), the one we use, was called the “installation cost” model by Panigrahi [28]. He gave a pseudo-polynomial time
algorithm for the minimum \( s-t \) path problem in this model; Cabello and Colin de Verdière [6] recently proved that this problem
is weakly NP-hard. Panigrahi et al. [29] gave an \( O(\log n) \)-approximation algorithm for the minimum spanning tree problem in
the installation cost model, and this was shown to be best possible assuming \( P \neq \text{NP} \) [28, 29].

Panigrahi [28] formulated a generalization of all these power requirements called “activation networks,” where there is a
general 0-1 function \( f \) on pairs of vertex powers and an edge \( uv \) is activated if \( f(p_u, p_v) = 1 \). In fact, his model is more general
in that it allows for different functions \( f \) for different pairs of vertices. However, the model has a significant restriction in that
the power values are constrained to a discrete set \( D \) whose size is assumed to be polynomially bounded in \( n \). Panigrahi gave
\( O(\log n) \) approximation algorithms for various network survivability problems in this setting, such as minimum spanning tree,
minimum Steiner forest, and 2-edge/vertex connectivity. These algorithms were improved in follow-up papers [2, 26]. For
further background, refer to the survey by Nutov on activation network design problems [27].

Finally, we mention that there is also an older notion of node-weighted optimization, for example [18], for vertex-weighted
graphs. For vertices \( u \) and \( v \) with weight \( w_u \) and \( w_v \), respectively, the requirement to activate the edge \( uv \) is that \( p_u \geq w_u \) and
\( p_v \geq w_v \).

1.2 | Organization

The remainder of our paper is organized as follows. Section 2 contains some alternative formulations of MSPEC. The barrier
coverage application is treated in Section 3. Our approximation scheme for MSPEC is in Section 5. This is preceded by Section
4 on a bottleneck version of the problem, and followed by Section 7 on speeding up the approximation algorithm. Some more
general and some restricted versions of MSPEC are considered in Section 6, and Section 8 concludes the paper.

2 | ALTERNATIVE FORMULATIONS OF MSPEC

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

In MSPEC we are searching for an edge cut. The set of edges in an \( s-t \) cut forms a bipartite graph. Given an edge cut, the
power assignment that will remove those edges corresponds to a minimum “\( w \)-vertex cover” ([32, Chapter 17]), that is, an
assignment of weights \( y_v \) to the vertices so that \( y_u + y_v \geq w_{uv} \) for all edges \( uv \) in the cut. In a bipartite graph the minimum \( w \)-vertex
cover is dual 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; for example, see Figure 2A.

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 \( \Pi_{st} \) be the set of all paths in \( G \) from \( s \) to \( t \). We have a nonnegative variable \( p_u \) for each \( u \in V \) and a 0-1
variable \( x_{uv} \) for each edge \( uv \in E \) with the intended interpretation that the cut edges have \( x_{uv} = 1 \).

\[
\min \sum_{u \in V} p_u \\
\text{s.t.} \sum_{uv \in \pi} x_{uv} \geq 1 \quad \forall \pi \in \Pi_{st} \\
p_u + p_v \geq w_{uv} x_{uv} \quad \forall uv \in E
\]
This integer programming formulation has an integrality gap of at least 2. Consider the graph shown in Figure 2B. 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 at least $\frac{1}{0.5} = 2$.

3 | BARRIER COVERAGE IN A SENSOR NETWORK

Our motivation for studying the MSPEC 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 modeled 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, that is, 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 an axis-parallel rectangle and the sensor network must detect any path that crosses the rectangle from below to above. Kumar et al. [20] 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 to the intersection graph of the disks augmented with two vertices, one for each of the vertical edges of the rectangle—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 3 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 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 MSPEC can be applied to compute minimum shrinkage for rectangular barrier coverage.

Our new measure is defined in Section 3.2, and the reduction to MSPEC is given in Section 3.3. We begin with further background on barrier coverage in Section 3.1.

3.1 | Background on barrier coverage

As mentioned above, Kumar et al. [20] 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 [5] applied this measure 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. Bereg and Kirkpatrick showed that thickness provides a constant-factor approximation to resilience. Korman et al. [19] provided fixed-parameter tractable algorithms (parameterized by the resilience) and a PTAS when each point of the plane is covered by few 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. [23] 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. [22] 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 [33] and Mitchell [24].

For further background refer to Wang’s survey on coverage problems in sensor networks [34] (in particular, Section 6).

### 3.2 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 \( s_i \), that is, 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 \( R_2 \), that is, there is a path from \( R_1 \) to \( R_2 \) in the free space between the shrunken disks. See Figure 1.

### 3.3 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 \( uv \) when the unit disks centered at \( u \) and \( v \) intersect, that is, the edge set is \( \{uv : u, v \in P \text{ and } \text{dist}(u, v) \leq 1\} \).

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 \( uv \) to be \( (2 - \text{dist}(u, v)) \). This is the amount of shrinkage at \( u \) and \( v \) that is needed to make the disks at \( u \) and \( v \) nonintersecting. 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 that extend the line segments forming the left and right boundaries of \( R \), respectively, and, for line \( L \), let \( \text{dist}(u, L) \) be the distance from point \( u \) to line \( L \). Define the weight of an edge \( su \) to be \( (1 - \text{dist}(u, R_l)) \). 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 \( tu \) to be \( (1 - \text{dist}(u, R_r)) \). It is straightforward to show that **MSPEC** on the resulting graph solves the minimum shrinkage problem where we interpret \( p_i \), as the amount to shrink the disk centered at \( v \) (Figure 4).

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

### 4 MINIMUM BOTTLENECK SHARED-POWER EDGE CUT

To design an FPTAS we need some approximation to the **MSPEC**. An easy approximation can be obtained by considering a closely related problem, which we call the “Minimum Bottleneck Shared-Power Edge Cut Problem”. This problem is similar to **MSPEC** in that we want to assign powers to the vertices such that if we remove every edge with weight smaller than the sum of the powers on its endpoints, then \( s \) and \( t \) become disconnected. 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 \( uv \), with \( u, v \in V \) and \( 2p \geq w_{uv} \), and the edges \( uv \), with \( u \in V, v \in \{s, t\} \) and \( p \geq w_{uv} \). See Figure 5.

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. [23]. 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. It is interesting that the problem can be solved in a more general nongeometric setting.

We will see that the minimum bottleneck shared-power edge cut can be found in near-linear time, and that it provides an \( n \)-approximation to the MSPEC problem. In Section 7 we will describe an alternative 2-approximation algorithm for MSPEC that can also be used to design an FPTAS—in fact, a faster FPTAS. We present the current method first because the minimum bottleneck shared-power edge cut problem is of independent interest, and the algorithm for it is efficient and simple.

**Theorem 1.** Minimum bottleneck shared-power edge cut in \( G \) can be solved optimally in the same time that is needed to find a minimum spanning tree in \( G \) (with respect to some modified weights).

**Proof.** Define the modified weight \( \tilde{w}_{uv} \) of edge \( uv \) to be \( \frac{1}{2} w_{uv} \) if \( u, v \in V \), or \( w_{uv} \) if \( u \in V \) and \( v \in \{s, t\} \). The modified weight represents the power requirement to remove an edge: if \( p \geq \tilde{w}_{uv} \), then the edge \( uv \) is removed. Let \( T \) be a maximum spanning tree of \( G \) with respect to the modified weights \( \tilde{w} \), let \( \pi \) be the \( s-t \) path in \( T \), and define \( p^* = \min \{ \tilde{w}_{uv} | uv \in \pi \} \).

Thus, \( p^* \) is the minimum modified weight of an edge in the \( s-t \) path of \( T \).

We claim that \( p^* \) is the optimum power. First note that if we choose \( p < p^* \), then no edge of the path \( \pi \) is removed, and thus the remaining graph has an \( s-t \) path. To show the other direction, we use the following standard property of maximum spanning trees, often called the cut property: for each partition \( A, B \) of the vertex set of \( G \), the edge cut \( E(A, B) = \{uv \in E | u \in A, v \in B \} \) contains an edge \( e \) of \( T \) such that \( \tilde{w}_e = \max \{ \tilde{w}_{uv} | uv \in E(A, B) \} \). Let \( e^* \) be an edge of \( \pi \) with \( w_{e^*} = p^* \). Removing the edge \( e^* \) from \( T \) separates the vertex set into two parts \( A \) and \( B \) with \( s \in A \) and \( t \in B \). From the cut-property we have that \( e^* \) has the heaviest modified weight of the edge cut \( E(A, B) \), that is, \( \tilde{w}_{e^*} = \max \{ \tilde{w}_{uv} | uv \in E(A, B) \} \). It follows that the power assignment \( p^* \) removes all the edges of \( E(A, B) \), and thus \( p^* \) is a feasible solution for the minimum bottleneck shared-power edge cut in \( G \).
Algorithm 1 Construction of $G'$

For each $v \in V$, make $c = \lceil n^2/e \rceil$ copies of $v$ numbered $v(0), v(1), \ldots, v(c-1)$. $V' = \{ v(i) \mid v \in V, 0 \leq i < c \} \cup \{ s(0), t(0) \}$

$\alpha = \epsilon p^*/n$

$E' = \{ u(i)v(j) \mid ia + ja < w_{uv} \} \cup \{ u(i)x(0) \mid x \in \{ s, t \}, ia < w_{ux} \}$

Algorithmically, we only have to compute a maximum spanning tree of $G$ with respect to the modified weights $\tilde{w}$, and then compute $p^*$ in linear time. Finding a maximum spanning tree or a minimum spanning tree are computationally equivalent problems: a maximum spanning tree with respect to weights $\tilde{w}_v$ is the same as a minimum spanning tree with respect to weights $-\tilde{w}_v$. If we want nonnegative weights we can equivalently use $c - \tilde{w}_v$, where $c$ is a large enough constant, for example $c = \max\{ \tilde{w}_{uv} \mid uv \in E \}$. □

Regarding the actual time complexity, note that finding a minimum spanning tree can be done in randomized linear time [17], in deterministic $O(m(a(m, n))$ time [8], where $\alpha$ is the (extremely slowly growing) functional inverse of Ackermann's function, or with an algorithm [30] that is asymptotically optimal, but whose running time is unknown.

5 APPROXIMATION SCHEME FOR MSPEC

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 $\frac{c}{n}$ 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. We obtain an easy bound from the minimum bottleneck shared-power edge cut. In Section 7 we describe how to obtain a 2-approximation algorithm, which eventually leads to a faster approximation scheme for MSPEC. 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 2. $p^* \leq OPT \leq np^*$.

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

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 $\alpha = \epsilon p^*/n$ for each vertex, then the total error over all vertices will be at most $\epsilon p^* \leq \epsilon OPT$. 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 $\alpha$. Since the total power that we might assign to the vertex is $np^*$, the number of copies of the vertex that we need is $c = \lceil np^*/\alpha \rceil = \lceil n^2/e \rceil$. We will replace each vertex of $V$ by a sequence of $c$ vertices, $v(0), v(1), \ldots, v(c-1)$. To handle all vertices in an unified way, it is convenient to use $s(0) = s$ and $t(0) = t$. We refer to $v(j)$ as a copy of $v$.

Removing the first $k$ vertices of the sequence will correspond to assigning power $ka$ to $v$. Note the “shift” in indices, which may be confusing: removing the vertices $v(0), v(1), \ldots, v(k-1)$ will correspond to assigning power $ka$ to $v$. Such indexing is useful for the edges. In $G'$ we will assign edges to the copies $v(i)$ to reflect this relation to the power assignment. More precisely, for $u, v \in V \cup \{ s, t \}$ we put an edge $u(i)v(j)$ in $G'$ if and only if $ia + ja < w_{uv} \cdot \alpha$. (Note that for $u \in V$ and $x \in \{ s, t \}$ we put the edge $u(i)x(0)$ in $G'$ if and only if $ia < w_{ux} \cdot \alpha$.) A related idea of discretizing the choices using vertices combined with minimum cuts was developed by Hochbaum et al. [14, 15] for integer linear programs with at most two variables per inequality. They rely on an algorithm by Picard [31] for finding the minimum-cost closure of a directed graph.

The precise construction of $G'$ is given in Algorithm 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

$k^*_v = \lfloor \frac{|\{j | v(j) \in K^* \}|}{|V|} \rfloor$ for each $v \in V$

$p_v = k^*_v \cdot a$ for each $v \in V$

return $p = (p_v)_{v \in V}$ as the solution for MSPEC on $G$

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 1. If $v \in V$ and $i_1 < i_2$, then in $G'$ the neighborhoods of $v(i_1)$ and $v(i_2)$ are related by $N_{G'}(v(i_1)) \supseteq N_{G'}(v(i_2))$.

Proof. We show that $v(i_2) u(j) \in E'$ implies $v(i_1) u(j) \in E'$. If the edge $v(i_2) u(j)$ is in $E'$, where $u \in V \cup \{s, t\}$, then $i_2 \alpha + ja < w_{uv}$, which implies that $i_1 \alpha + ja < w_{uv}$ and thus the edge $v(i_1) u(j)$ is in $E'$.

Claim 2. The copies of $v$ in $K^*$ are $v(0), v(1), \ldots, v(k^*_v - 1)$.

Proof. We 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 1, $N(v(i_1)) \supseteq N(v(i_2))$. 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 3. A solution computed by Algorithm 2 is a feasible solution for MSPEC and the sum of the assigned powers is $a |K^*|$.\[\sum_{v \in V} p_v = \sum_{v \in V} k^*_v \cdot a = a \sum_{v \in V} k^*_v = a |K^*| \]

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 $p^* = (p^*_v)_{v \in V}$ be an optimum power assignment for MSPEC, and let $p^*_v = p^*_v = 0$.

Lemma 4. $G'$ has a vertex cut $K$ of size $\sum \lceil \frac{p^*_v}{a} \rceil$. Thus $|K| \leq \sum \lceil \frac{p^*_v}{a} \rceil$.
is not a vertex of \( G' - K \). Thus no copy \( u(i)v(j) \) of the edge \( uv \) exists in \( G' - K \). This proves that \( K \) is an \( s(0)-t(0) \) vertex cut in \( G' \) and completes the proof of the lemma.

**Theorem 5.** Algorithm 2 FPTAS for the MSPEC problem. Furthermore, the running time of Algorithm 2 is \( O(n^{5.5} m \varepsilon^{-2.5}) \).

**Proof.** By Lemma 3 we know that Algorithm 2 gives a solution to MSPEC of cost \( \alpha |K^*| \), and by Lemma 4 we have \( |K^*| \leq \sum \left\lceil \frac{p^*_v}{\alpha} \right\rceil \). Therefore

\[
\text{Approximate solution} = \alpha |K^*| \leq \sum_{v \in V} \left\lceil \frac{p^*_v}{\alpha} \right\rceil \cdot \alpha < \sum_{v \in V} \left( \frac{p^*_v}{\alpha} + 1 \right) \cdot \alpha = \sum_{v \in V} (p^*_v + \alpha) = \sum_{v \in V} p^*_v + na = OPT + na.
\]

Thus the approximation ratio is at most

\[
\frac{OPT + na}{OPT} = \frac{OPT + \varepsilon p^*}{OPT} = 1 + \frac{\varepsilon p^*}{OPT} \leq (1 + \varepsilon),
\]

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 \( O(n^{12/7} m) = O(n^{5.5}) \) for a graph with \( n \) vertices and \( m \) edges using a modified version of Diniz’s flow algorithm ([25], Chapter 1, [32], Corollary 9.7a). If our original graph \( G \) has \( n \) vertices and \( m \) edges, then the graph \( G' \) has \( |V'| = nc = O\left(\frac{n^2}{\varepsilon}\right) \) vertices and at most \( |E'| \leq |E|^2 = O\left(\frac{mn^2}{\varepsilon^2}\right) \) edges, and Algorithm 2 finds a vertex cut in graph \( G' \) in \( O(n^{5.5} m \varepsilon^{-2.5}) \) time.

**5.1 Geometric interpretation for the minimum shrinkage**

In Section 3.3 we have shown how the minimum shrinkage problem for rectangular barrier coverage can be formulated as an MSPEC problem. The algorithm discussed in the current section has a simple geometric interpretation in this setting. We are replacing each unit disk (radius 1) by a sequence of \( \varepsilon \) concentric disks of radii \( 1, 1-\alpha, 1-2\alpha, \ldots, 1-(c-1)\alpha \). (Here the disks have to be open, that is, without their boundary.) The graph \( G' \) is the intersection graph of these open disks, with the additional vertices \( s \) and \( t \) representing the vertical edges of the rectangular domain. This graph \( G' \) is precisely the graph that has to be considered for the original barrier coverage problem of Kumar et al. [20] for this augmented set of disks, and the solution to the barrier coverage problem is given by a minimum \( s-t \) cut in \( G' \).

**6 VARIATIONS OF MSPEC**

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

**6.1 Uniform/integral edge weights**

We use the alternative formulation of MSPEC given at the start of Section 2. 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 König’s theorem and its generalization to weights (Egerváry’s theorem, see [32], Theorem 17.1) 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 \( O(n^{12/7} m) \) 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 \( O((nW)^{12/7}(mW^2)) = O(n^{12/7}mW^{52}) \). Thus it provides a pseudo polynomial time algorithm for MSPEC with integer edge weights.
6.2 | Polynomial domain

In activation network design problems, one of the most common assumptions is that of a “polynomial domain” for the power values [26, 28]. 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 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 \( 0 \notin D^v \) for all \( v \). We first define a bottleneck version of MSPEC.

Theorem 6. Minimum bottleneck shared-power edge cut with vertex cuts in \( G \) can be solved optimally in the same time that is needed to find a minimum spanning tree in \( G \) (with respect to some modified weights).

Proof. Define the modified weight \( \tilde{w}_{uv} \) of edge \( uv \) to be \( \frac{c_v}{c_u + c_v} w_{uv} \) if \( u, v \in V \), or \( c_u w_{uv} \) if \( u \in V \) and \( v \in \{s, t\} \). The modified weight represents the power requirement to remove an edge: if \( k \geq \tilde{w}_{uv} \), then the edge \( uv \) is removed.

Let \( T \) be a maximum spanning tree of \( G \) with respect to the modified weights \( \tilde{w} \), let \( \pi \) be the \( s \rightarrow t \) path in \( T \), and define \( k^* = \min \{ \tilde{w}_{uv} | uv \in \pi \} \). Thus, \( k^* \) is the minimum modified weight of an edge in the \( s \rightarrow t \) path of \( T \).

As in the proof of Theorem 1, we can prove that \( k^* \) is the optimum power.

Furthermore, if \( OPT \) is the optimal solution of MSPEC with vertex costs then we can prove the following analogue of Lemma 2.

Lemma 7. \( k^* \leq OPT \leq nk^* \).
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 $\left\lceil \frac{n^2}{\epsilon} \right\rceil$ copies of every vertex in $V$. We set $\alpha$ equal to $\frac{\epsilon k^2}{n}$ and define the edge set $E' = \{(u(i), v(j)) \mid \frac{in_{u(i)}}{\alpha} + \frac{in_{v(j)}}{\alpha} < w_u, v \} \cup \{(u(i), x) \mid x \in \{s, t\}, \frac{in_{u(i)}}{\alpha} < w_u, x \}$. The remainder of the algorithm and the proofs carry over after similar modifications.

7. FASTER APPROXIMATION SCHEME FOR MSPEC

In this section, we explain how to speed up our FPTAS from the current running time of $O(n^{2.5}m \epsilon^{-2.5})$ to $O(n^3m \epsilon^{-2.5} + m^3/\log m)$. To accomplish this, we discretize the problem using a 2-approximation for MSPEC instead of the $n$-approximation given by the minimum bottleneck shared power edge cut. Let $OPT$ be the optimum solution value for MSPEC.

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 6.2 solves this discrete version in $O((\sum \text{deg}(v))/3)\log(\sum \text{deg}(v)) = O(m^3/\log m)$ time.

We now use the fact that the discrete MSPEC provides a 2-approximation. This was proved more generally by Panigrahi [28, Lemma 3.11], but we include his short argument.

**Theorem 8.** [28]. Discrete MSPEC is an 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 $p^* = (p^*_e)_{e \in V}$. For each edge $uv \in C$, select the endpoint $v$ of higher power in $p^*$. Then $p^*_e \geq \frac{1}{2}w_{uv}$. For each vertex $v$ that is selected by some edge of $C$, we raise $p^*_v$ to the maximum $w_{uv}$ over all edges $uv$ of $C$ that select $v$. With this procedure, we at most double the power of each vertex $v$. Furthermore, any vertex not selected can have its power decreased to 0. The new power values still activate all edges of $C$, provide a solution to discrete MSPEC, and use at most double power than $p^*$.

Let $Z$ be the optimal value of a solution to the discrete MSPEC problem on graph $G$. Clearly $OPT \leq Z$, and from Theorem 8 we have $Z \leq 2OPT$. Thus

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

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

$$O(|V'|^{1/2}|E'|) = O((n \cdot n/\epsilon)^{1/2}(m \cdot (n/\epsilon)^2)) = O(n^3m \epsilon^{-2.5}).$$

Since the time to compute $Z$ is $O(m^3/\log m)$, the overall running time of our fast FPTAS is $O(n^3m \epsilon^{-2.5} + m^3/\log m)$. In addition to improving the running time, this improves the space complexity by at least $O(n^2)$. We summarize.

**Theorem 9.** There is a polynomial time approximation scheme (FPTAS) for the MSPEC problem with running time $O(n^3m \epsilon^{-2.5} + m^3/\log m)$.

8. CONCLUSIONS AND OPEN PROBLEMS

The most interesting open question is to settle the complexity (in P or NP-complete) of the two problems that we solved by means of an FPTAS, namely the MSPEC problem and the special case of minimum shrinkage for a rectangular barrier.

After our paper was written the first author and Éric Colin de Verdière [6] showed that the general minimum shrinkage problem (without the rectangle restriction) is weakly NP-hard. They also proved weak NP-hardness of a problem that is in some sense dual to MSPEC, namely the “minimum activation path” problem where the activated edges must contain a path from vertex $s$ to vertex $t$.

Although their proof techniques do not seem to carry over, we conjecture that MSPEC is also weakly NP-hard.

Turning to other measures of barrier coverage, we note that NP-hardness of the basic 0-1 barrier coverage measure of Kumar et al. [20] (without the rectangle restriction) is still open after more than a decade.
Another direction for future research is to measure barrier coverage using a nonlinear model of sensor deterioration over distance (e.g., the Elphes sensing model, see [12]).

ACKNOWLEDGMENTS
This work was begun at the Fifth Annual Workshop on Geometry and Graphs, held at the Bellairs Research Institute in Barbados, March 5-10, 2017. We thank the organizers and all participants for the productive and positive atmosphere. We thank Lap Chi Lau for helpful suggestions.

ORCID
Kshitij Jain 🐘 https://orcid.org/0000-0002-5515-132X
Anna Lubiw 🐘 https://orcid.org/0000-0002-2338-361X

REFERENCES
[1] P.K. Agarwal, S. Har-Peled, H. Kaplan, and M. Sharir, Union of random Minkowski sums and network vulnerability analysis, Discrete Comput. Geom. 52 (2014), 551–582. https://doi.org/10.1007/s00454-014-9626-1.
[2] H.M. Alqahtani and T. Erlebach, Approximation algorithms for disjoint st-paths with minimum activation cost, Proceedings of the International Conference on Algorithms and Complexity (CIAC), Springer, 2013, pp. 1–12. https://doi.org/10.1007/978-3-642-38233-8_1.
[3] E. Althaus, G. Calinescu, I.I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky, Power efficient range assignment for symmetric connectivity in static ad hoc wireless networks, Wirel. Netw. 12 (2006), 287–299. https://doi.org/10.1007/s11276-005-5275-x.
[4] E. Angel, E. Bampis, V. Chau, and A. Kononov, Min-power covering problems, Proceedings of the International Symposium on Algorithms and Computation (ISAAC), Springer, 2015, pp. 367–377. https://doi.org/10.1007/978-3-662-48971-0_32.
[5] S. Bereg and D. Kirkpatrick, “Approximating barrier resilience in wireless sensor networks,” Algorithmic Aspects of Wireless Sensor Networks, S. Dolev (ed.) Springer-Verlag, 2009, pp. 29–40. https://doi.org/10.1007/978-3-642-05434-1_5.
[6] S. Cabello and E. Colin de Verdière, Hardness of minimum barrier shrinkage and minimum activation path, arXiv preprint arXiv:1910.04228, 2019. URL: http://arxiv.org/abs/1910.04228
[7] S. Cabello, K. Jain, A. Lubiw, and D. Mondal, Minimum shared-power edge cut, CoRR, abs/1806.04742, 2018. URL: http://arxiv.org/abs/1806.04742, arXiv:1806.04742.
[8] B. Chazelle, A minimum spanning tree algorithm with inverse-Ackermann type complexity, J. ACM 47 (2000), 1028–1047. https://doi.org/10.1145/355541.355562.
[9] J. Cheriyan, T. Hagerup, and K. Mehlhorn, An o(n)-time maximum-flow algorithm, SIAM J. Comput. 25 (1996), 1144–1170. https://doi.org/10.1137/S0097539793251876.
[10] N. Cohen and Z. Nutov, Approximating minimum power edge-multi-covers, J. Combin. Optim. 30 (2015), 563–578. https://doi.org/10.1007/s10878-013-9652-6.
[11] H.N. Djidjev. Approximation algorithms for computing minimum exposure paths in a sensor field, ACM Trans. Sensor Netw. (TOSN) 7 (2010), 23. https://doi.org/10.1145/1807048.1807052.
[12] R. Elhabyan, W. Shi, and M. St-Hilaire, Coverage protocols for wireless sensor networks: Review and future directions, J. Commun. Netw. 17 (2019). https://doi.org/10.1109/JCN.2019.000005.
[13] M.T. Hajiaghayi, G. Kortsarz, V.S. Mirrokni, and Z. Nutov, Power optimization for connectivity problems, Math. Programming 110 (2007), 195–208. https://doi.org/10.1007/s10107-006-0057-5.
[14] D.S. Hochbaum, N. Megiddo, J. (Sefli) Naor, and A. Tamir, Tight bounds and 2-approximation algorithms for integer programs with two variables per inequality, Math. Programming 62 (1993), 69–83. https://doi.org/10.1007/BF01585160.
[15] D.S. Hochbaum and J. (Sefli) Naor, Simple and fast algorithms for linear and integer programs with two variables per inequality, SIAM J. Comput. 23 (1994), 1179–1192. https://doi.org/10.1137/S0097539793251876.
[16] K. Jain, Minimum shared-power edge cut, Master’s thesis, University of Waterloo, Canada, 2018. URL: http://hdl.handle.net/10012/13664
[17] D.R. Karger, P.N. Klein, and R.E. Tarjan, A randomized linear-time algorithm to find minimum spanning trees, J. ACM 42 (1995), 321–328. https://doi.org/10.1145/201019.201022.
[18] P. Klein and R. Ravi, A nearly best-possible approximation algorithm for node-weighted Steiner trees, J. Algorithms 19 (1995), 104–115. https://doi.org/10.1006/jagm.1995.1029.
[19] M. Korman, M. Löffler, R.J. Silveira, and D. Strash, On the complexity of barrier resilience for fat regions and bounded ply, Comput. Geom. 72 (2018), 34–51. https://doi.org/10.1016/j.comgeo.2018.02.006.
[20] S. Kumar, T.H. Lai, and A. Arora, Barrier coverage with wireless sensors, Wirel. Netw. 13 (2007), 817–834. https://doi.org/10.1007/s11276-006-9856-0.
[21] Y. Lando and Z. Nutov, On minimum power connectivity problems, J. Discrete Algorithms 8 (2010), 164–173. https://doi.org/10.1016/j.jda.2009.03.002.
[22] M. Meunier, F. Koushanfar, Q. Gang, G. Veltri, and M. Potkonjak, Exposure in wireless sensor networks: Theory and practical solutions, Wirel. Netw. 8 (2002), 433–454. https://doi.org/10.1023/A:1016586011473.
[23] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, Exposure in wireless ad-hoc sensor networks, Proceedings of the 7th Annual International Conference on Mobile Computing and Networking (MobiCom), ACM, 2001, pp. 139–150. https://doi.org/10.1145/381677.381691.
[24] J.S.B. Mitchell, On maximum flows in polyhedral domains, J. Comput. Syst. Sci. 40 (1990), 88–123. https://doi.org/10.1016/0022-0000(90)90020-L.
[25] H. Nagamochi and T. Ibaraki, Algorithmic Aspects of Graph Connectivity, Cambridge University Press, Cambridge, 2008.
[26] Z. Nutov, Survivable network activation problems, Theoret. Comput. Sci. 514 (2013), 105–115. https://doi.org/10.1016/j.tcs.2012.10.054.
How to cite this article: Cabello S, Jain K, Lubiw A, Mondal D. Minimum shared-power edge cut. Networks. 2020;75:321–333. https://doi.org/10.1002/net.21928