eprints@whiterose.ac.uk
https://eprints.whiterose.ac.uk/

On Sparse Hitting Sets:
From Fair Vertex Cover to Highway Dimension

Johannes Blum ✉ 🏠 🔘
Universität Konstanz, Germany

Yann Disser ✉ 🔘
Technische Universität Darmstadt, Germany

Andreas Emil Feldmann ✉ 🏠 🔘
Charles University, Prague, Czechia

Siddharth Gupta ✉ 🏠 🔘
University of Warwick, Coventry, UK

Anna Zych-Pawlewicz ✉ 🔘
University of Warsaw, Poland

──── **Abstract** ────

We consider the Sparse Hitting Set (Sparse-HS) problem, where we are given a set system $(V, \mathcal{F}, \mathcal{B})$ with two families $\mathcal{F}, \mathcal{B}$ of subsets of the universe V. The task is to find a hitting set for $\mathcal{F}$ that minimizes the maximum number of elements in any of the sets of $\mathcal{B}$. This generalizes several problems that have been studied in the literature. Our focus is on determining the complexity of some of these special cases of Sparse-HS with respect to the *sparseness* k, which is the optimum number of hitting set elements in any set of $\mathcal{B}$ (i.e., the value of the objective function).

For the Sparse Vertex Cover (Sparse-VC) problem, the universe is given by the vertex set V of a graph, and $\mathcal{F}$ is its edge set. We prove NP-hardness for sparseness $k \geq 2$ and polynomial time solvability for $k = 1$. We also provide a polynomial-time 2-approximation algorithm for any k. A special case of Sparse-VC is Fair Vertex Cover (Fair-VC), where the family $\mathcal{B}$ is given by vertex neighbourhoods. For this problem it was open whether it is FPT (or even XP) parameterized by the sparseness k. We answer this question in the negative, by proving NP-hardness for constant k. We also provide a polynomial-time $(2 - \frac{1}{k})$-approximation algorithm for Fair-VC, which is better than any approximation algorithm possible for Sparse-VC or the Vertex Cover problem (under the Unique Games Conjecture).

We then switch to a different set of problems derived from Sparse-HS related to the *highway dimension*, which is a graph parameter modelling transportation networks. In recent years a growing literature has shown interesting algorithms for graphs of low highway dimension. To exploit the structure of such graphs, most of them compute solutions to the r-Shortest Path Cover (r-SPC) problem, where $r > 0$, $\mathcal{F}$ contains all shortest paths of length between r and $2r$, and $\mathcal{B}$ contains all balls of radius $2r$. It is known that there is an XP algorithm that computes solutions to r-SPC of sparseness at most h if the input graph has highway dimension h. However it was not known whether a corresponding FPT algorithm exists as well. We prove that r-SPC and also the related r-Highway Dimension (r-HD) problem, which can be used to formally define the highway dimension of a graph, are both W[1]-hard. Furthermore, by the result of Abraham et al. [ICALP 2011] there is a polynomial-time $O(\log k)$-approximation algorithm for r-HD, but for r-SPC such an algorithm is not known. We prove that r-SPC admits a polynomial-time $O(\log n)$-approximation algorithm.

1 Introduction

In this paper, we study the problem of finding a sparse hitting set. That is, we are given a set system $(V, \mathcal{F}, \mathcal{B})$ on universe V with two set families $\mathcal{F}, \mathcal{B} \subseteq 2^V$, and a feasible solution is a set $H \subseteq V$ that hits (i.e., intersects) every set of $\mathcal{F}$. Instead of minimizing the overall size of the solution however, we think of the sets of $\mathcal{B}$ as being small and we would like to distribute the solution H among the sets in $\mathcal{B}$ as evenly as possible. Intuitively and depending on the context, the sets in $\mathcal{B}$ are balls in some metric and the hitting set should be sparse within them. That is, we want to find a hitting set for $\mathcal{F}$ that minimizes the largest intersection with the sets of $\mathcal{B}$. Formally, the SPARSE HITTING SET (SPARSE-HS) problem with input $(V, \mathcal{F}, \mathcal{B})$ is defined by the following integer linear program (ILP) with indicator variables x_v for each $v \in V$ encoding membership in the solution $H \subseteq V$.

$$\min k \text{ such that:} \quad \sum_{v \in F} x_v \geq 1 \qquad \forall F \in \mathcal{F} \qquad \text{(SPARSE-HS-ILP)}$$

$$\sum_{v \in B} x_v \leq k \qquad \forall B \in \mathcal{B}$$

$$x_v \in \{0, 1\} \qquad \forall v \in V$$

The SPARSE-HS problem generalizes several problems studied in the literature, with applications in for instance cellular [24], communication [25], and transportation [2, 22] networks. Our aim in this paper is to determine the complexity of some basic variants of SPARSE-HS, and we are specifically interested in the complexity depending on the *sparseness*, which is the solution value k of (SPARSE-HS-ILP). In general, SPARSE-HS contains the HITTING SET problem by setting $\mathcal{B} = \{V\}$, and thus does not admit any $g(k)$-approximation in $f(k) \cdot n^{O(1)}$ time [21], for any computable functions f and g, where $n = |V|$, under ETH.

Sparse Vertex Cover. A much easier special case of HITTING SET is the well-known VERTEX COVER problem: for the SPARSE VERTEX COVER (SPARSE-VC) problem the set system is given by a graph $G = (V, E)$ so that $\mathcal{F} = E$ and $\mathcal{B} \subseteq 2^V$. We show that this problem is NP-hard for any $k \geq 2$, even on very simple input graphs.

▶ **Theorem 1.** *SPARSE-VC is* NP*-hard for any $k \geq 2$, even if the input graph is a matching.*

Note that this hardness result implies that, unless P=NP, SPARSE-VC does not admit an *XP algorithm* with runtime $n^{f(k)}$ for any function f (the problem is paraNP-hard parameterized by the sparseness k). This is in contrast to the VERTEX COVER problem, which is known to be *fixed-parameter tractable (FPT)* parameterized by the solution size s, which means that it can be solved much more efficiently in $f(s) \cdot n^{O(1)}$ time for some function f (which can be shown [10] to be 1.2738^s). On the other hand, we will show that for $k = 1$ the SPARSE-VC problem is polynomial-time solvable, which together with the previous hardness result settles the complexity of the SPARSE-VC problem for every sparseness value k.

▶ **Theorem 2.** *Sparse-VC is polynomial time solvable for $k = 1$.*

As we will see, Theorem 1 also implies that Sparse-VC does not admit a polynomial-time $(3/2 - \varepsilon)$-approximation algorithm for any $\varepsilon > 0$, unless P=NP. Moreover, As Vertex Cover is a special case of Sparse-VC with $\mathcal{B} = \{V\}$, any polynomial time $(2 - \varepsilon)$-approximation algorithm for Sparse-VC would refute the Unique Games Conjecture (UGC) [28]. On the positive side, we show that we can match this conditional approximation lower bound with a 2-approximation algorithm. This means that Sparse-VC can be approximated as well as the Vertex Cover problem, which also admits a 2-approximation [28] in polynomial time.

▶ **Theorem 3.** *Sparse-VC admits a polynomial time 2-approximation algorithm.*

Fair Vertex Cover. A special case of Sparse-VC is the Fair Vertex Cover (Fair-VC) problem where the family of sets $\mathcal{B}$ is given by closed neighbourhoods, i.e., if $N[v]$ is the set containing vertex v and all neighbours of v in G then $\mathcal{B} = \{N[v] \mid v \in V\}$ (alternatively, $\mathcal{B}$ contains all balls of radius 1). The fairness constraint was introduced by Lin and Sahni [25] in the context of communication networks, and has since then been studied for several types of problems (cf. Section 1.1), including Vertex Cover [22, 18, 26]. In contrast to this paper, in [22, 18] the problem is defined slightly differently by considering open neighbourhoods, i.e., $\mathcal{B} = \{N[v] \setminus \{v\} \mid v \in V\}$, and we call this version Open-Fair-VC. Notably, the parameterized complexity of Open-Fair-VC has been studied for a plethora of parameters, including treedepth, treewidth, feedback vertex set, modular width [22], and the total solution size $|H|$ [18], and most of these results also apply to Fair-VC with closed neighbourhoods.

Jacob et al. [18] observe that it is NP-hard to decide if a vertex cover of size s exists, if every neighbourhood is allowed to only contain at most k vertices of the solution H, for a given constant $k \geq 3$: this follows from the fact that Vertex Cover is NP-hard on sub-cubic graphs [17]. While the authors of [18] call this problem Fair Vertex Cover as well, note that this is significantly different from the Fair-VC problem studied in this paper as well as the Open-Fair-VC problem studied in [22]. In particular, on sub-cubic graphs both of these problems as defined here always trivially have a solution for $k \geq 3$,[1] and thus the NP-hardness of Vertex Cover on sub-cubic graphs does not immediately imply NP-hardness of Fair-VC or Open-Fair-VC. In fact, for the natural parameterization by the sparseness k the complexity of Open-Fair-VC (and also Fair-VC) has so far been unknown.[2] We answer this open problem by showing NP-hardness of Fair-VC for $k \geq 3$ and of Open-Fair-VC for $k \geq 4$ on more complex input graphs when compared to Sparse-VC.

▶ **Theorem 4.** *Fair-VC is NP-hard for any $k \geq 3$ and Open-Fair-VC is NP-hard for any $k \geq 4$, even on planar input graphs.*

Thus, as for Sparse-VC, we can conclude that Fair-VC and Open-Fair-VC do not admit XP algorithms parameterized by k. For the cases when $k \leq 2$, Jacob et al. [18] provide a polynomial time algorithm that solves their version of Fair Vertex Cover, which however also works for the Fair-VC and Open-Fair-VC problems as defined in this paper. Hence this settles the complexity of Fair-VC for every value of k, and only leaves the value $k = 3$ open for Open-Fair-VC.

---

[1] Observe that it is never necessary to pick a vertex v and all its neighbours.
[2] Tomáš Masařík, personal communication.

In terms of approximation, interestingly we are able to obtain a slightly better algorithm for FAIR-VC than for SPARSE-VC, namely a $(2 - \frac{1}{k})$-approximation. This beats the best possible approximation for SPARSE-VC and VERTEX COVER under UGC [28]. In particular, the following theorem implies that for the smallest value $k = 3$ for which FAIR-VC is NP-hard, we can obtain a solution of sparseness 5. We leave open whether a solution of sparseness 4 can be computed in polynomial time for FAIR-VC if $k = 3$, and whether better approximation algorithms are possible for OPEN-FAIR-VC.

▶ **Theorem 5.** *FAIR-VC admits a polynomial time $(2 - \frac{1}{k})$-approximation algorithm.*

Shortest Path Cover and Highway Dimension.   We now turn to a different set of problems derived from SPARSE-HS, which as we shall see generalize FAIR-VC. Given a value $r > 0$ and an edge-weighted graph G, for the r-SHORTEST PATH COVER (r-SPC) problem the family $\mathcal{F}$ is given by shortest paths of length between r and $2r$ and the family $\mathcal{B}$ is given by balls of radius $2r$. That is, let $\mathcal{P}_r$ contain $S \subseteq V$ if and only if S is the vertex set of a path in G, which is a shortest path (according to the edge weights) and whose length is in the range $(r, 2r]$. Furthermore, let $\mathrm{dist}(u, v)$ be the length of a shortest u-v-path and let $B_r(v) = \{u \in V \mid \mathrm{dist}(u, v) \leq r\}$ denote the ball of radius r centered at v. Then for the r-SPC problem, $\mathcal{F} = \mathcal{P}_r$ and $\mathcal{B} = \{B_{2r}(v) \mid v \in V\}$.

The r-SPC problem finds applications in the context of the *highway dimension*, which is a graph parameter introduced by Abraham et al. [2] to model transportation networks. To define the highway dimension, we define a problem related to r-SPC called r-HIGHWAY DIMENSION (r-HD), where for each vertex $v \in V$ the task is to find a hitting set for all shortest paths of length in $(r, 2r]$ intersecting the ball $B_{2r}(v)$, and we need to minimize the largest such hitting set. Note that compared to r-SPC the quantification is reversed, i.e., for r-SPC there is a hitting set that is small in every ball, while for r-HD for every ball there is a small hitting set (thus r-HD is not a special case of SPARSE-HS). The *highway dimension* of an edge-weighted graph G is the smallest integer h such that there is a solution to r-HD of value at most h in G for every $r > 0$.

There is empirical evidence [4] that road networks have small highway dimension, and it has been conjectured [13] that public transportation networks (especially those stemming from airplane networks) have small highway dimension as well.[3] Therefore, there has been some effort to devise algorithms [13, 2, 12, 14, 11, 11, 5, 19, 8, 7] for problems on low highway dimension graphs that naturally arise in transportation networks. It is known [1] that if the highway dimension of a graph G is h, then the r-SPC problem on G has sparseness at most h for every $r > 0$, but not vice versa, as the sparseness of r-SPC can be much smaller than h. However, since a solution to r-SPC consists of one hitting set $H \subseteq V$ for the whole graph, it is more convenient to work with algorithmically than the n hitting sets for all balls of radius $2r$ that form a solution to r-HD. Therefore, algorithms exploiting the structure of graphs of low highway dimension typically compute a solution to the r-SPC problem for each of the $O(n^2)$ relevant values of r given by the pairwise distances between vertices.

For graphs of low highway dimension, Abraham et al. [1] give an algorithm that for each relevant value of r computes a solution to the r-HD problem, in order to obtain a solution to r-SPC with sparseness at most the value of the r-HD solution. While Abraham et al. [1] propose to use an approximation algorithm for r-HD (see below), note that the

---

[3] In fact there are several definitions of the highway dimension, with the one presented here being well-suited for public transportation networks, cf. [13, 6]

r-HD problem admits an XP algorithm with runtime $n^{O(k)}$, since for any ball $B_{2r}(v)$ it can construct the set system given by all shortest paths of length in $(r, 2r]$ intersecting $B_{2r}(v)$, for which it can then try every possible k-tuple of vertices as a solution. This algorithm can thus be used to compute solutions to r-SPC of sparseness at most h in $n^{O(h)}$ time if the input graph has highway dimension h. Interestingly, it is not possible to compute solutions of optimum sparseness for r-SPC using an XP algorithm due to the NP-hardness of FAIR-VC: consider an r-SPC instance with unit edge weights and value $r = 1/2$. Since every edge is a shortest path between its endpoints, the r-SPC problem on this instance is equivalent to FAIR-VC. As argued above however, no XP algorithm exists for FAIR-VC, unless P=NP.

In light of the growing amount of work on problems on low highway dimension graphs, it would be very useful to have a faster algorithm to solve r-HD in order to compute a hitting set for r-SPC of corresponding sparseness. While it is known that computing the highway dimension is NP-hard [13] and this also implies that r-HD is NP-hard, r-HD might still be FPT and allow algorithms with runtime $f(k) \cdot n^{O(1)}$ for some function f. However, we will show that it is unlikely that such algorithms exist. In particular, we prove that r-HD is W[1]-hard parameterized by the solution value k. We also prove that r-SPC does not admit FPT algorithms (in particular, k here denotes the optimum sparseness and not just an upper bound that we would obtain by solving r-HD, as suggested above). While already the above reduction from FAIR-VC to r-SPC excludes FPT algorithms for r-SPC, this only excludes such algorithms for very small values of r, in fact the smallest relevant value for r (as the problem becomes trivial for even smaller values). A priori it is not clear whether r-SPC admits FPT (or XP) algorithms for large values of r. In our reduction however, the value of r takes the largest relevant value, so that there exists a ball of radius $2r$ containing the whole graph.

▶ **Theorem 6.** *Both r-HD and r-SPC are W[1]-hard parameterized by their solution values k, where $2r$ is the radius of the input graph.*

One caveat of this hardness result is that it does *not* answer the question of whether computing the highway dimension is FPT or not. This is because the presented reduction only shows hardness of r-HD for a large value r. However, for smaller values of r the solution value to r-HD is unbounded in the constructed graph, and thus the graph does not have bounded highway dimension. This means that it might still be possible to compute the highway dimension in FPT time, but not using the existing tools provided by Abraham et al. [1], where each value r is considered separately. Instead, if such an algorithm exists it must consider the structure of the whole graph. We leave open whether there is such an algorithm.

As mentioned above, Abraham et al. [1] propose an approximation algorithm for r-HD: under the assumption that all shortest paths are unique (which can always be achieved by slightly perturbing the edge lengths), r-HD admits a polynomial time $O(\log k)$-approximation algorithm. Due to the fact that the sparseness of r-SPC can be a lot smaller than the solution value to r-HD,[4] it is not known how to obtain such an algorithm for r-SPC. However, we prove the existence of a weaker $O(\log n)$-approximation algorithm.

▶ **Theorem 7.** *r-SPC admits a polynomial time $O(\log n)$-approximation algorithm.*

---

[4] as for instance witnessed by the graphs constructed in the reduction for Theorem 4 and value $r = 1/2$.

Dense Matching.   Finally, in light of the above results for SPARSE-VC, we also consider the dual DENSE MATCHING problem, where we are given a graph $G = (V, E)$ and the task is to find a matching $M \subseteq E$ maximizing the smallest number of matching edges induced by a set in the family $\mathcal{B}$, i.e., the minimum $|M \cap E(B)|$ over all $B \in \mathcal{B}$, where $E(B) = \{\{u, v\} \in E \mid u, v \in B\}$. Despite the MAXIMUM MATCHING problem being polynomial-time solvable, we show that DENSE MATCHING does not admit a polynomial time $(2 - \varepsilon)$-approximation, even if $\mathcal{B}$ is restricted to balls of radius two, unless P=NP. Interestingly, a matching 2-approximation seems a lot harder to come by compared to SPARSE-VC, and we leave open whether a constant approximation is possible for DENSE MATCHING.

▶ **Theorem 8.** *It is NP-hard to approximate DENSE MATCHING within $2 - \varepsilon$ for any $\varepsilon > 0$, even if $\mathcal{B} = \{B_2(v) \mid v \in V\}$ where all edges have weight 1.*

Due to space restrictions, the proof of Theorem 8 can be found in the appendix.

1.1   Related Work

Apart from the work cited above, we here list some additional related work. Kanesh et al. [20] study the FAIR FEEDBACK VERTEX SET problem, where the family $\mathcal{F}$ contains all vertex sets of cycles of the input graph (in this case $\mathcal{F}$ is not part of the input). They prove results on the parameterized complexity of several versions of this problem, where the considered parameters include treewidth, treedepth, neighbourhood diversity, the total solution size, and the maximum vertex degree. Jacob et al. [18] consider the parameterized complexity of the FAIR SET and FAIR INDEPENDENT SET problems, but also Π-FAIR VERTEX DELETION, where Π is any property expressible in first order (FO) logic. Knop et al. [22] study Π-FAIR VERTEX DELETION for properties Π expressible in monadic second order (MSO_1) logic parameterized by the twin cover number. They also consider FAIR-VC parameterized by treedepth, feedback vertex number, and modular width. Agrawal et al. [3] study the parameterized complexity of the MINIMUM MEMBERSHIP DOMINATING SET problem, where $\mathcal{F} = \mathcal{B} = \{N[v] \mid v \in V\}$, and consider parameterizations by pathwidth, sparseness, and vertex cover number.

While in this paper we study SPARSE-HS problems on graphs where the universe is the set of vertices, another line of work studies variants of SPARSE-HS when the universe is the edge set. For instance, the work of Lin and Sahni [25] that introduced the fairness constraint, studies the FAIR FEEDBACK EDGE SET problem, where the family $\mathcal{F}$ contains the edge sets of all cycles of the input graph. Masařík and Toufar [26] consider the parameterized complexity of the Π-FAIR EDGE DELETION problem, where Π is a property expressible in FO logic or in MSO logic. For each of these problems they study parameterizations by the treewidth, pathwidth, treedepth, feedback vertex set number, neighbourhood diversity, and vertex cover number. Kolman et al. [23] study the Π-FAIR EDGE DELETION problem on graphs of bounded treewidth, where Π is any property expressible in MSO logic. They also give tight polynomial-time $O(\sqrt{n})$-approximation algorithms for FAIR ODD CYCLE TRANSVERSAL and FAIR MIN CUT, where the family $\mathcal{F}$ contains all edge sets of odd cycles and (s, t)-paths for given vertices s and t, respectively. Another notable problem is MIN DEGREE SPANNING TREE, where the family $\mathcal{F}$ consists of every edge cut under the fairness constraint. Fürer and Raghavachari [15] prove that the problem is NP-hard but a solution of sparseness $k + 1$ can be computed in polynomial-time.

Regarding the complexity of computing the highway dimension, it is interesting to note that Abraham et al. [1] show that any set system given by unique shortest paths has VC-dimension 2 (this observation also leads to the above mentioned $O(\log k)$-approximation

algorithm for r-HD). At the same time, Bringmann et al. [9] prove that the Hitting Set problem is W[1]-hard for set systems of VC-dimension 2. Hence it is intriguing to think that the latter reduction could possibly be modified to also prove W[1]-hardness for r-HD or r-SPC. However, it seems that shortest paths exhibit a lot more structure than general set systems of VC-dimension 2, and thus it is unclear how to obtain a hardness result for r-HD or r-SPC based on [9]. Instead, a more careful reduction as provided in Theorem 6 seems necessary.

2   Sparse Vertex Cover

In this section we consider the Sparse-VC problem and start by proving NP-hardness for any $k \geq 2$.

▶ **Theorem 1.** *Sparse-VC is* NP-*hard for any $k \geq 2$, even if the input graph is a matching.*

Proof. We reduce from a variant of the satisfiability problem called Exactly-3-Sat, meaning that all clauses contain exactly three literals. This problem was shown to be NP-complete in [16]. For a set of variables $X = \{x_i\}_i$, we use the notation $\bar{X} := \{\bar{x}_i\}_i$.

Let an instance of Exactly-3-Sat be given by a set of variables $X = \{x_i\}_{i=1,\ldots,n}$ and a set of clauses $\mathcal{C} = \{C_j\}_{j=1,\ldots,m}$ with $C_j \subset X \cup \bar{X}, |C_j| = 3$. We define the graph $G = (V, E)$ by $V = X \cup \bar{X} \cup \{y_i, \bar{y}_i \mid i \in \{1, \ldots, k-1\}\}$ and $E = \{\{x_i, \bar{x}_i\} \mid i \in \{1, \ldots, n\}\} \cup \{\{y_i, \bar{y}_i\} \mid i \in \{1, \ldots, k-1\}\}$. We further let $\tilde{C} = C \cup \{y_1, \bar{y}_1, \ldots, y_{k-2}, \bar{y}_{k-2}\}$ and choose $\mathcal{B} = \{\{x_i, \bar{x}_i, y_1, \bar{y}_1, \ldots, y_{k-1}, \bar{y}_{k-1}\} \mid i \in \{1, \ldots, n\}\} \cup \{\tilde{C} \mid C \in \mathcal{C}\}$. This construction can be carried out in linear time and G is a matching. For NP-hardness, it remains to show that G has a vertex cover $H \subseteq V$ satisfying $|H \cap B| \leq k$ for every $B \in \mathcal{B}$ if and only if the given Exactly-3-Sat instance has a satisfying assignment.

To see this, first assume that the given Exactly-3-Sat instance has a satisfying assignment $\alpha\colon X \to \{0,1\}$ and extend α to $\bar{X}$ by letting $\alpha(\bar{x}) := 1 - \alpha(x)$. We construct the vertex cover $H = \{x \in X \cup \bar{X} \mid \alpha(x) = 0\} \cup \{y_1, \ldots, y_{k-1}\}$. Indeed, H is a vertex cover, since every edge of G is covered by exactly one of its endpoints. It also follows that, for every $i \in \{1, \ldots, n\}$, we have $|\{x_i, \bar{x}_i, y_1, \bar{y}_1, \ldots, y_{k-1}, \bar{y}_{k-1}\} \cap H| = k$. By definition of α, for every $C \in \mathcal{C}$, we have $\sum_{x \in C} \alpha(x) \geq 1$, hence $|H \cap \tilde{C}| = |H \cap C| + k - 2 = \sum_{x \in C} \alpha(\bar{x}) + k - 2 = 3 - \sum_{x \in C} \alpha(x) + k - 2 \leq k$.

Conversely, suppose that there exists a vertex cover $H \subseteq V$ with $|H \cap B| \leq k$ for all $B \in \mathcal{B}$. We claim that $\alpha(x) = |\{\bar{x}\} \cap H|$ defines a satisfying assignment for the given Exactly-3-Sat instance. Observe that we must have $|\{x_i, \bar{x}_i\} \cap H| \geq 1$ for all $i \in \{1, \ldots, n\}$ and $|\{y_i, \bar{y}_i\} \cap H| \geq 1$ for all $i \in \{1, \ldots, k-1\}$, since H needs to cover all edges. Since $\{x_i, \bar{x}_i, y_1, \bar{y}_1, \ldots, y_{k-1}, \bar{y}_{k-1}\} \in \mathcal{B}$, it follows that $|\{x_i, \bar{x}_i, y_1, \bar{y}_1, \ldots, y_{k-1}, \bar{y}_{k-1}\} \cap H| \leq k$ for all $i \in \{1, \ldots, n\}$. Together, we obtain $|\{x_i, \bar{x}_i\} \cap H| = 1$ for all $i \in \{1, \ldots, n\}$. We can therefore extend α to $\bar{X}$ by setting $\alpha(\bar{x}) = 1 - \alpha(x) = |\{x\} \cap H|$. Finally, for $C \in \mathcal{C}$, we have $\tilde{C} \in \mathcal{B}$ and thus $|H \cap \tilde{C}| \leq k$ and moreover $|H \cap \tilde{C}| \geq |H \cap C| + k - 2$, which implies $|H \cap C| \leq 2$. It follows that $\sum_{x \in C} \alpha(x) = 3 - \sum_{x \in C} \alpha(\bar{x}) = 3 - \sum_{x \in C} |\{x\} \cap U| = 3 - |H \cap C| \geq 1$, thus α is a satisfying assignment.   ◀

We can observe that Theorem 1 also shows that Sparse-VC does not admit a $(3/2 - \varepsilon)$-approximation algorithm for any $\varepsilon > 0$, unless P=NP. This follows from the fact that for $k = 2$, such an algorithm would be able to determine whether a given instance of Sparse-VC admits a solution of sparseness $2 \cdot (3/2 - \varepsilon) < 3$, i.e., of optimal sparseness 2.

Let us now consider the Sparse-VC problem for sparseness $k = 1$. We show that in this case, Sparse-VC can be reduced to the 2-SAT problem, which is commonly known to admit a linear time algorithm. This yields the following theorem.

▶ **Theorem 2.** *Sparse-VC is polynomial time solvable for $k = 1$.*

Proof. The instance of the Sparse-VC problem is given by a graph $G = (V, E)$ and a set of balls $\mathcal{B} \subseteq 2^V$. Given this instance, we construct a 2-Sat formula ϕ, which is solvable if and only if the Sparse-VC instance has a solution. Moreover, we can reconstruct the fair vertex cover for $(V, E, \mathcal{B})$ given a satisfying assignment to ϕ.

To construct ϕ, we first assign a variable x_v to each vertex $v \in V$. Next, for every edge $\{u, v\} \in E$ we create a clause $(x_u \vee x_v)$ and add it to ϕ, so that we are guaranteed that any satisfying assignment will correspond to a valid vertex cover. Now we have to enforce, that for each ball $B \in \mathcal{B}$, at most one variable in the set $\{x_v\}_{v \in \mathcal{B}}$ is set to true. This is done by adding $\binom{|B|}{2}$ clauses: for each pair $v, u \in B, u \neq v$ we add a clause $(\bar{x}_v \vee \bar{x}_u)$ to enforce that x_v and x_u cannot be both true. In this way, we ensure that only one variable of the set $\{x_v\}_{v \in \mathcal{B}}$ is set to true. Thus, the final formula ϕ takes the following form:

$$\phi = \bigwedge_{\{u,v\} \in E} (x_u \vee x_v) \wedge \bigwedge_{B \in \mathcal{B}, u,v \in B : u \neq v} (\bar{x}_v \vee \bar{x}_u)$$

Given a satisfying assignment for ϕ, we reconstruct the solution to $(V, E, \mathcal{B})$ by taking the vertices whose variable was set to true. We already argued that such a solution is feasible for Sparse-VC with $k = 1$. In the opposite direction, given a solution to Sparse-VC with $k = 1$, we find an assignment by setting the variables corresponding to the vertices of the solution to true: the clauses corresponding to edges are then satisfied due to the solution being a vertex cover, and the remaining clauses corresponding to $\mathcal{B}$ are satisfied because the solution picks at most one vertex from each $B \in \mathcal{B}$.   ◀

Finally, we show how to obtain a 2-approximation algorithm for Sparse-VC. This approximation factor is optimal unless the Unique Games Conjecture fails, as Vertex Cover is a special case of Sparse-VC with $\mathcal{B} = \{V\}$.

▶ **Theorem 3.** *Sparse-VC admits a polynomial time 2-approximation algorithm.*

Proof. We consider the relaxation of (Sparse-HS-ILP) for a given graph $G = (V, E)$:

$$\min k \text{ such that:} \quad x_u + x_v \geq 1 \qquad\qquad \forall uv \in E \qquad\qquad (1)$$

$$\sum_{v \in B} x_v \leq k \qquad\qquad \forall B \in \mathcal{B} \qquad\qquad (2)$$

$$x_v \geq 0 \qquad\qquad \forall v \in V \qquad\qquad (3)$$

Note that in any feasible solution to this LP, for any edge $\{u, v\}$ at least one of the two variables x_u and x_v has value at least $1/2$, due to constraints (1) and (3). Thus the set $W = \{v \in V \mid x_v \geq 1/2\}$ of all vertices with value at least $1/2$, is a vertex cover for the input graph. The sparseness of this solution can be bounded using (2) for any set $B \in \mathcal{B}$:

$$|W \cap B| \leq 2 \sum_{v \in B} x_v \leq 2k$$

Thus solving the above LP relaxation optimally in polynomial time and then outputting the set W, gives a 2-approximation algorithm for Sparse-VC.   ◀
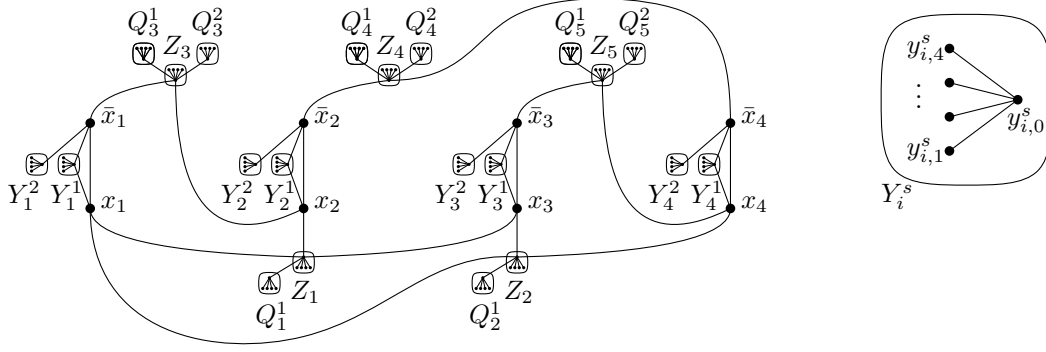
Figure 1 Left: The graph G for the formula $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_3 \vee x_4) \wedge (\bar{x}_1 \vee x_2) \wedge (\bar{x}_2 \vee \bar{x}_4) \wedge (\bar{x}_3 \vee x_4)$ and $k = 4$. Right: A star Y_i^s with center $y_{i,0}^s$ and leaves $y_{i,1}^s, \ldots, y_{i,4}^s$. Similarly, Z_j and Q_j^s denote stars with centers $z_{j,0}$ and $q_{j,0}^s$, and leaves $z_{j,1} \ldots, z_{j,4}$ and $q_{j,1}^s, \ldots, q_{j,4}^s$, respectively.

3 Fair Vertex Cover

Let us now consider the (OPEN-)FAIR-VC problem, where the balls $\mathcal{B}$ are given by (open) vertex neighborhoods. We first show NP-hardness of FAIR-VC and OPEN-FAIR-VC for $k \geq 3$ and $k \geq 4$, respectively.

▶ **Theorem 4.** *FAIR-VC is NP-hard for any $k \geq 3$ and OPEN-FAIR-VC is NP-hard for any $k \geq 4$, even on planar input graphs.*

Proof. We reduce from the PLANAR 2P1N-3-SAT problem. In this variant of satisfiability, all clauses contain two or three literals, and we may assume that every variable appears exactly twice as a positive literal and exactly once as a negative literal over all clauses. In addition, we may assume that the bipartite graph connecting clauses to the variables they contain is planar. This variant of satisfiability was shown to be NP-complete in [27].

We first consider the FAIR-VC problem and later show how to modify our reduction for OPEN-FAIR-VC. Let an instance of PLANAR 2P1N-3-SAT be given by a set of variables $X = \{x_i\}_{i=1,\ldots,n}$ and a set of clauses $\mathcal{C} = \{C_j\}_{j=1,\ldots,m}$ with $C_j \subset X \cup \bar{X}$, $|C_j| \in \{2, 3\}$. We define the graph $G = (V, E)$ by

$$V = \bigcup_{i=1}^{n} \left(\{x_i, \bar{x}_i\} \cup \bigcup_{s=1}^{k-2} \bigcup_{r=0}^{k} \{y_{i,r}^s\} \right) \cup \bigcup_{j=1}^{m} \left(\bigcup_{r=0}^{k} \{z_{j,r}\} \cup \bigcup_{s=1}^{k-|C_j|} \bigcup_{r=0}^{k} \{q_{j,r}^s\} \right)$$

and

$$E = \bigcup_{i=1}^{n} \left(\{\{x_i, \bar{x}_i\}\} \cup \bigcup_{s=1}^{k-3} \{\{x_i, y_{i,0}^s\}\} \cup \bigcup_{s=1}^{k-2} \{\{\bar{x}_i, y_{i,0}^s\}\} \cup \bigcup_{s=1}^{k-2} \bigcup_{r=1}^{k} \{\{y_{i,0}^s, y_{i,r}^s\}\} \right)$$

$$\cup \bigcup_{j=1}^{m} \left(\bigcup_{x \in C_j} \{\{x, z_{j,0}\}\} \cup \bigcup_{r=1}^{k} \{\{z_{j,0}, z_{j,r}\}\} \cup \bigcup_{s=1}^{k-|C_j|} \left(\{z_{j,0}, q_{j,0}^s\} \cup \bigcup_{r=1}^{k} \{q_{j,0}^s, q_{j,r}^s\} \right) \right).$$

This construction (illustrated in Figure 1) can be carried out in linear time and G is planar.

For NP-hardness of FAIR-VC, it remains to show that G has a vertex cover $H \subset V$ satisfying $|H \cap N[v]| \leq k$ for every $v \in V$ if and only if the given PLANAR 2P1N-3-SAT instance has a satisfying assignment.

To see this, first assume that the given PLANAR 2P1N-3-SAT instance has a satisfying assignment $\alpha \colon X \to \{0,1\}$ and extend α to $\bar{X}$ by letting $\alpha(\bar{x}) := 1 - \alpha(x)$. We construct the vertex cover $H = \{x \in X \cup \bar{X} \mid \alpha(x) = 0\} \cup \bigcup_{i=1}^{n} \bigcup_{s=1}^{k-2} \{y_{i,0}^s\} \cup \bigcup_{j=1}^{m} \left(\{z_{j,0}\} \cup \bigcup_{s=1}^{k-|Cj|} \{q_{j,0}^s\} \right)$. Indeed, H is a vertex cover, since all edges are of the form $\{x, \bar{x}\}$ or incident to some $y_{i,0}^s$, $z_{j,0}$, or $q_{j,0}^s$. Also, $|H \cap N[v]| = 1$ for $v \in \bigcup_{i=1}^{n} \bigcup_{s=1}^{k-2} \bigcup_{r=1}^{k} \{y_{i,r}^s\} \cup \bigcup_{j=1}^{m} \bigcup_{r=1}^{k} \{z_{j,r}\} \cup \bigcup_{j=1}^{m} \bigcup_{s=1}^{k-|C_j|} \bigcup_{r=1}^{k} \{q_{j,r}^s\}$, $|H \cap N[v]| = 2$ for $v \in \bigcup_{i=1}^{n} \bigcup_{s=1}^{k-2} \{y_{i,0}^s\} \cup \bigcup_{j=1}^{m} \bigcup_{s=1}^{k-|C_j|} \{q_{j,0}^s\}$, and $|H \cap N[v]| = k$ for $v \in X \cup \bar{X}$. It remains to consider $v = z_{j,0}$ for some $j \in \{1, \ldots, m\}$. Observe that $\sum_{x \in C_j} \alpha(x) \geq 1$ since α is a satisfying assignment. It holds that $|H \cap N[z_{j,0}]| = 1 + k - |C_j| + \sum_{x \in C_j} |\{x\} \cap H| = 1 + k - |C_j| + \sum_{x \in C_j} (1 - \alpha(x)) \leq 1 + k - |C_j| + |C_j| - 1 = k$. In either case, we conclude that $|H \cap N[v]| \leq k$.

Conversely, suppose that there exists a vertex cover $H \subseteq V$ with $|H \cap N[v]| \leq k$ for all $v \in V$. Let $i \in \{1, \ldots, n\}$ and $s \in \{1, \ldots, k-2\}$. Since H is a vertex cover with $|H \cap N[y_{i,0}^s]| \leq k$ and $\deg(y_{i,0}^s) > k$, we must have $y_{i,0}^s \in H$. Similarly, we must have $z_{j,0} \in H$ for all $j \in \{1, \ldots, m\}$ and $q_{j,0}^s \in H$ for all $j \in \{1, \ldots, m\}, s \in \{1, \ldots, k-|C_j|\}$. We claim that $\alpha(x) = |\{\bar{x}\} \cap H|$ defines a satisfying assignment for the given PLANAR 2P1N-3-SAT instance. To see this, first observe that $|\{x, \bar{x}\} \cap H| = 1$ for all $x \in X$ since H is a vertex cover and $|H \cap N[x]| \leq k$. We can therefore extend α to $\bar{X}$ by setting $\alpha(\bar{x}) = 1 - \alpha(x) = |\{x\} \cap H|$. Recall that for for all $j \in \{1, \ldots, m\}$ we have $|H \cap N[z_{j,0}]| \leq k$ and $\left\{ z_{j,0}, q_{j,0}^1, \ldots, q_{j,0}^{k-|C_j|} \right\} \subseteq H$, which implies $\left| H \cap \left(N[z_{j,0} \setminus \left\{ z_{j,0}, q_{j,0}^1, \ldots, q_{j,0}^{k-|C_j|} \right\} \right) \right| \leq k - (1 + k - |C_j|) = |C_j| - 1$. With this, for $j \in \{1, \ldots, m\}$, we have $\sum_{x \in C_j} \alpha(x) = |C_j| - \sum_{x \in C_j} \alpha(\bar{x}) = |C_j| - \sum_{x \in C_j} |\{x\} \cap H| \geq |C_j| - \left| \left(N[z_{j,0}] \setminus \left\{ z_{j,0}, q_{j,0}^1, \ldots, q_{j,0}^{k-|C_j|} \right\} \right) \cap H \right| \geq |C_j| - (|C_j| - 1) = 1$. We conclude that α is a satisfying assignment.

We now turn to OPEN-FAIR-VC and modify the above reduction as follows. The first difference is that there is now only one Y_i^s gadget for each variable x_i, so we call it Y_i. The second difference is that Y_i is different from Y_i^s from the previous reduction, because now Y_i is responsible for picking only one vertex among $\{x_i, \bar{x}_i\}$ to the solution. To be more precise Y_i is now a star with a center $y_{i,0}$ connected to $y_{i,1}, \ldots y_{i,k-1}$, but now also each $y_{i,j}, j \in \{1 \ldots, k-1\}$ is a center of a star, connected to $y_{i,1}^j, \ldots, y_{i,k}^j$. In other words, Y_i is a tree of depth two rooted at $y_{i,0}$, where the root has $k-1$ children and each child of the root has k children. In the constructed graph, for each variable x_i both literals x_i and $\bar{x}_i$ are now connected to $y_{i,0}$ instead of $y_{i,0}^s$ vertices from the previous reduction. The remaining part of the graph is precisely the same as in the reduction for Theorem 1.

Assume we have an OPEN-FAIR-VC solution for the constructed graph and the given parameter $k \geq 4$. Observe, that for each i the vertices $y_{i,0}$ and $y_{i,1} \ldots y_{i,k-1}$ must be taken to the solution since their degree is $k+1$. Therefore, since vertex $y_{i,0}$ has $k-1$ neighbors other than x_i and $\bar{x}_i$, only one vertex among x_i and $\bar{x}_i$ can be taken to the solution. The remaining part of the argument is the same as in the proof for FAIR-VC: we construct a satisfying assignment by setting this literal to false, whose corresponding vertex was taken to the solution. Similarly as before, each clause has at most two negated literals. The opposite direction is analogous. ◀

The previous reduction actually also shows that for any $\varepsilon > 0$, it is NP-hard to compute a $(4/3 - \varepsilon)$-approximation for FAIR-VC, as for $k = 3$, a $(4/3 - \varepsilon)$-approximate solution actually has sparseness 3. Still, we are able to compute a $(2 - \frac{1}{k})$-approximation for FAIR-VC, which is slightly better than our result for SPARSE-VC and also better then the best possible approximation ratio for VERTEX COVER (and thus SPARSE-VC) under UGC. In particular, our algorithm implies that for the smallest value $k = 3$ for which FAIR-VC is NP-hard, we

can obtain a solution of sparseness 5. We leave open whether a solution of sparseness 4 can be computed in polynomial time for FAIR-VC if $k = 3$, and whether better approximation algorithms are possible for OPEN-FAIR-VC.

▶ **Theorem 5.** *FAIR-VC admits a polynomial time $(2 - \frac{1}{k})$-approximation algorithm.*

Proof. As for the 2-approximation algorithm for SPARSE-VC (cf. Theorem 3), we consider the relaxation of (SPARSE-HS-ILP). However, in order to improve the approximation ratio, observe that in any solution of cost k to FAIR-VC, every vertex of degree more than k must be contained in the solution (otherwise some edge incident to such a vertex is not covered). Thus we may guess the optimum sparseness $k^\star$, define the set of high degree vertices $D = \{v \in V \mid \deg(v) > k^\star\}$ and add the constraint $x_v = 1$ for every $v \in D$ to the above LP relaxation. We again let W be the set of vertices with value at least $1/2$, which is a feasible vertex cover. If for any closed neighbourhood $N[v] \in \mathcal{B}$ we have $N[v] \subseteq W$ then all neighbours of v are contained in W, and thus we may remove v from W and still obtain a vertex cover. We repeat this iteratively for each vertex until we obtain a vertex cover W for which no closed neighbourhood is entirely contained in W. In particular, for any $v \notin D$ we have $|W \cap N[v]| \le k^\star$. For $v \in D$ on the other hand, since $x_v = 1$ we get

$$|W \cap N[v]| \le |D \cap N[v]| + 2 \sum_{u \in N[v] \setminus D} x_u \le (2x_v - 1) + 2 \sum_{u \in N[v] \setminus \{v\}} x_u \le 2k - 1.$$

This means that the set W yields a $(2 - \frac{1}{k})$-approximation.   ◀

4   Hardness of Highway Dimension and Shortest Path Cover

In this section, we study the parameterized complexity of r-HD and r-SPC, and show the following theorem.

▶ **Theorem 6.** *Both r-HD and r-SPC are W[1]-hard parameterized by their solution values k, where $2r$ is the radius of the input graph.*

To prove this, we present a parameterized reduction from CLIQUE to r-HD. This reduction also shows W[1]-hardness for r-SPC, as the constructed graph G has radius at most $2r$, i.e., any solution for r-HD is also a solution for r-SPC of the same cost and vice versa.

Let $H = (V, E)$ be a graph and let $k \in \mathbb{N}$. Denote the number of vertices and edges of H by n and m, respectively. For convenience we treat H as a bidirected graph, i.e. we replace every edge $\{u, v\} \in E$ with directed edges (u, v) and (v, u). Let C be a constant whose value will be determined later on. We construct a graph G such that r-HD has a solution of value $k' = 4Ck(k-1) + \binom{k}{2} + k + 3$ on G for $r = 2^m$ if and only if H contains a clique if size k. In the following, we call the individual elements of a solution for r-HD also *hubs*.

The graph G contains $k(k-1)$ gadgets: For all $1 \le i, j \le k$ satisfying $i \ne j$ there is a gadget $G_{i,j}$. Choosing a certain set of hubs from $G_{i,j}$ means that $G_{i,j}$ represents a pair (w_i, w_j) of adjacent vertices of H. The idea of the reduction is to have a pair (w_i, w_j) from every $G_{i,j}$ such that

(i) if $G_{i,j}$ represents (x, y), then $G_{j,i}$ represents (y, x), and

(ii) if $G_{i,j}$ represents (x, y) and $G_{i,j'}$ represents (x', y'), then $x = x'$.

If these two conditions are fulfilled, it follows that there are k distinct vertices $w_1, \ldots, w_k$ which are pairwise adjacent, i.e. $\{w_1, \ldots, w_k\}$ is a clique of size k.
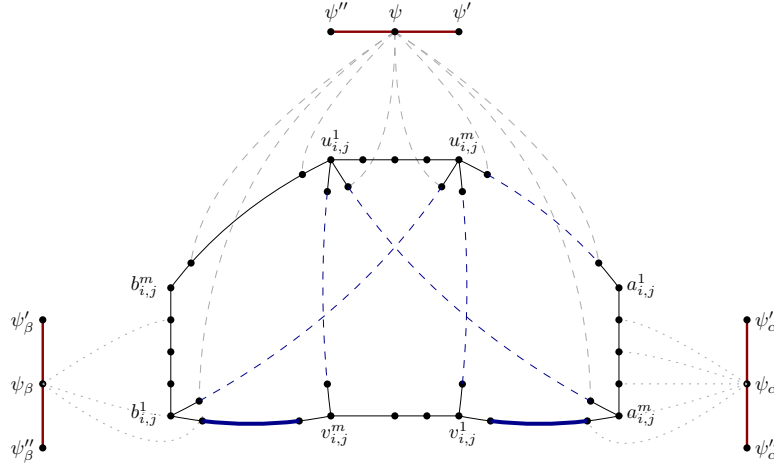
■ **Figure 2** A gadget $G_{i,j}$ and the global vertices. Solid black edges have length 1, dashed blue edges have length $r - m + 1$, thick blue edges have length $r - 2m + 2$, dotted gray edges have length $m - 1$, dashed gray edges have length $r/2$ and red edges have length r.

Every gadget $G_{i,j}$ contains a path $u_{i,j}^1, \ldots, u_{i,j}^m$, a path $v_{i,j}^1, \ldots, v_{i,j}^m$, a path $a_{i,j}^1, \ldots, a_{i,j}^m$, and a path $b_{i,j}^1, \ldots, b_{i,j}^m$, each consisting of $m-1$ edges of length 1. We identify every vertex of these paths with a pair (x, y) of adjacent vertices in H as follows: Fix any ordering $\prec$ on V and denote the resulting lexicographic ordering on $V \times V$ also by $\prec$. Define $\tau \colon E \to \{1, \ldots, m\}$ as

$$\tau(x, y) = |\{(u, v) \in E \mid (u, v) \prec (x, y)\}| + 1,$$

i.e. (x, y) is the $\tau(x, y)$-th edge according to $\prec$. This allows us for instance to associate the vertex $u_{i,j}^{\tau(x,y)}$ of $G_{i,j}$ with the edge (x, y) of H.

The four paths are connected as follows. For $z \in \{a, v, b\}$ we connect $u_{i,j}^m$ with $z_{i,j}^1$ and $z_{i,j}^m$ with $u_{i,j}^1$, each through a path of length $r - m + 3$. To that end we introduce vertices $u_{i,j}^{0,z}, u_{i,j}^{m+1,z}, z_{i,j}^{0,u}$ and $z_{i,j}^{m+1,u}$, and add the edges $\{u_{i,j}^1, u_{i,j}^{0,z}\}, \{u_{i,j}^m, u_{i,j}^{m+1,z}\}, \{z_{i,j}^1, z_{i,j}^{0,u}\}$, $\{z_{i,j}^m, z_{i,j}^{m+1,u}\}$ of length 1 and the edges $\{u_{i,j}^{m+1,z}, z_{i,j}^{0,u}\}, \{z_{i,j}^{m+1,u}, u_{i,j}^{0,z}\}$ of length $r - m + 1$.

Moreover, we add vertices $a_{i,j}^{m+1,v}, v_{i,j}^{0,a}, v_{i,j}^{m+1,b}, b_{i,j}^{0,v}$ and add edges $\{a_{i,j}^m, a_{i,j}^{m+1,v}\}, \{v_{i,j}^1, v_{i,j}^{0,a}\}, \{v_{i,j}^m, v_{i,j}^{m+1,b}\}, \{b_{i,j}^1, b_{i,j}^{0,v}\}$ of length 1 and edges $\{a_{i,j}^{m+1}, v_{0,a}\}, \{v_{i,j}^{m+1,b}, b_{i,j}^{0,v}\}$ of length $r - 2m + 2$. This is illustrated in Figure 2.

The idea is that the shortest path from $u_{j,j}^1$ to any of $a_{i,j}^{0,u}, v_{i,j}^{0,u}$, and $b_{i,j}^{0,u}$ has length $r + 1$ and that we will have to choose some pair (x, y) in order to hit these shortest paths through the hub $u_{i,j}^{\tau(x,y)}$. Still, the shortest paths between $a_{i,j}^{0,u}$ and $b_{i,j}^{0,u}$ and between $a_{i,j}^{m+1,u}$ and $b_{i,j}^{m+1,u}$ both have length $2r - 2m + 4 > r$, but are not hit, if we choose, e.g., the hub $u_{i,j}^2$. Hence, we introduce a shorter path between $a_{i,j}^{0,u}$ and $b_{i,j}^{0,u}$ and between $a_{i,j}^{m+1,u}$ and $b_{i,j}^{m+1,u}$, which will be hit by a global dummy hub: We add vertices ψ, ψ', ψ'' and the edges $\{\psi', \psi\}$ and $\{\psi'', \psi\}$, both of length r. Moreover, we add edges between ψ and $a_{i,j}^{0,u}, a_{i,j}^{m+1,u}, b_{i,j}^{0,u}$, and $b_{i,j}^{m+1,u}$, each of length $r/2$. The shortest $a_{i,j}^{0,u}$-$b_{i,j}^{0,u}$- and $a_{i,j}^{m+1,u}$-$b_{i,j}^{m+1,u}$-paths now have length r and pass through ψ. Furthermore, the shortest ψ'-ψ''-path has length $2r$ and we may assume w.l.o.g. that it is hit through the hub ψ.

We will show that if the final graph G admits a solution of value k', then there is a hitting set for $\mathcal{P}_r$ containing four vertices from every gadget $G_{i,j}$, which represent a pair (x, y) of adjacent vertices of H. Our construction needs to ensure that for these pairs (x, y) conditions i and ii are fulfilled. First we create C copies $G_{i,j}^1, \ldots, G_{i,j}^C$ of the graph $G_{i,j}$. For simplicity,

we confuse the graphs $G_{i,j}$ and $G_{i,j}^\lambda$ for $\lambda \in \{1, \dots, C\}$ when the context is clear. Our final construction will yield that if $G_{i,j}^\lambda$ represents (x, y) and $G_{i,j}^{\lambda'}$ represents (x', y'), then we have $(x, y) = (x', y')$. Note that the vertices ψ, ψ', and ψ'' are not part of any gadget $G_{i,j}$ and hence, we do not create copies of them.

For condition i we have to synchronise the gadgets $G_{i,j}$ and $G_{j,i}$. To that end, for all $1 \le i < j \le k$ and all $(x, y) \in E$ we add a vertex $\alpha_{i,j}^{(x,y)}$. Moreover, we add edges of weight m from (all C copies of) $a_{i,j}^{\tau(x,y)+1}$ and from (all C copies of) $a_{j,i}^{\tau(y,x)+1}$ to $\alpha_{i,j}^{(x,y)}$. This is illustrated in the appendix (Figure 3). The idea is that all shortest paths between $G_{i,j}$ and $G_{j,i}$ contained in $\mathcal{P}_r$ can be hit with one additional hub $\alpha_{i,j}^{(x,y)}$ if both gadgets agree on the pairs (x, y) and (y, x).

Still, the newly added edges of length m add new shortest paths to $\mathcal{P}_r$. For instance, in any $G_{i,j}$, the shortest path between $u_{i,j}^{m+1,a}$ and $\alpha_{i,j}^{\tau^{-1}(1)}$ has length $r + 2$. To ensure that it suffices to choose only $u_{i,j}^{\tau(x,y)}, v_{i_j}^{\tau(x,y)}$, and $\alpha_{i,j}^{(x,y)}$ as hubs, we remove these paths from $\mathcal{P}_r$ by creating a new shortest path between $u_{i,j}^{m+1,a}$ and $\alpha_{i,j}^{\tau^{-1}(1)}$, which passes through the dummy hub ψ. To that end, for all $1 \le i, j \le k$ satisfying $i \ne j$ we add an edge between ψ and $u_{i,j}^{0,a}, u_{i,j}^{m+1,a}$ and all $\alpha_{i,j}^{(x,y)}$, each of length $r/2$.

Similarly, we avoid "undesired" hubs covering shortest paths across different gadgets $G_{i,j}$ and $G_{j,i}$ by introducing new vertices $\psi_\alpha, \psi'_\alpha$ and ψ''_α and adding the edges $\{\psi'_\alpha, \psi_\alpha\}$ and $\{\psi''_\alpha, \psi_\alpha\}$ of length r and an edge of length $m - 1$ between ψ_α and all $a_{i,j}^{\tau(x,y)+1}$.

To fulfill condition ii we have to synchronise the gadget $G_{i,j}$ with every other gadget $G_{i,j'}$. To that end, for all $1 \le i \le k$ and all $x \in V$ we add a vertex β_i^x. Let $y_0, \dots, y_d$ be the neighbors of x such that $y_0 \prec \cdots \prec y_d$. For $1 \le i, j \le k, i \ne j$ we add an edge of weight $m + d$ between β_i^x and every (copy of) $b_{i,j}^{\tau(x,y_0)-1}$. Here the idea is that if two gadgets $G_{i,j}$ and $G_{i,j'}$ represent pairs (x, y) and (x', y') such that $x = x'$, then choosing β_i^x as a hub suffices to hit all relevant shortest paths between the two gadgets.

Again, we have to take care of newly created shortest paths. Therefore we add an edge of length $r/2$ between ψ and $u_{i,j}^{0,b}, u_{i,j}^{m+1,b}$ and all β_i^x. Moreover we handle shortest paths across different gadgets $G_{i,j}$ and $G_{i,j'}$ by introducing new vertices ψ_β, ψ'_β and ψ''_β and adding the edges $\{\psi'_\beta, \psi_\beta\}$ and $\{\psi''_\beta, \psi_\beta\}$ of length r and an edge of length $m + d - 1$ between ψ_β and every $b_{i,j}^{\tau(x,y_d)-1}$ where y_d is the maximum neighbor of x according to $\prec$. This concludes the construction of the graph G, which is also illustrated in the appendix (Figure 4).

We now show several properties of the graph G which allow us to prove Theorem 6. The following Lemma states that choosing four hubs from some gadget $G_{i,j}$ means that the gadget represents a unique pair (x, y). A proof can be found in the appendix.

▶ **Lemma 9.** *Let $1 \le i, j \le k$ where $i \ne j$ and let $\mathcal{H}_{i,j}$ be a hitting set for all shortest paths from $\mathcal{P}_r$ that are contained in $G_{i,j}$. It holds that $|\mathcal{H}_{i,j}| \ge 4$ and moreover, if $|\mathcal{H}_{i,j}| = 4$, then $G_{i,j}$ represents some (x, y), that is $\mathcal{H}_{i,j} = \left\{ u_{i,j}^{\tau(x,y)}, a_{i,j}^{\tau(x,y)}, v_{i,j}^{\tau(x,y)}, b_{i,j}^{\tau(x,y)} \right\}$.*

Moreover, we show that if a gadget $G_{i,j}$ represents some pair (x, y), then two certain shortest paths are not hit by the hubs of $G_{i,j}$. To that end, for any $(x, y) \in E$ and all $1 \le i < j \le k$ and $\lambda \in \{1, \dots, C\}$, let $A_{i,j}^{(x,y),\lambda}$ be the shortest path between $\alpha_{i,j}^{(x,y)}$ and the λ-th copy of $v_{i,j}^{\tau(x,y)-1}$. The length of $A_{i,j}^{(x,y),\lambda}$ is

$$\text{dist}(\alpha_{i,j}^{(x,y)}, a_{i,j}^{\tau(x,y)+1}) + \text{dist}(a_{i,j}^{\tau(x,y)+1}, a_{i,j}^{m+1,v}) + \text{dist}(a_{i,j}^{m+1,v}, v_{i,j}^{0,a}) + \text{dist}(v_{i,j}^{0,a}, v_{i,j}^{\tau(x,y)-1}) =$$
$$m + m - \tau(x,y) + r - 2m + 2 + \tau(x,y) - 1 = r + 1.$$

Similarly, we define $B_{i,j}^{x,\lambda}$ as the shortest path between β_i^x and the λ-th copy of $v_{i,j}^{\tau(x,y_d)+1}$, where y_d is the maximum neighbor of x. The path $B_{i,j}^{x,\lambda}$ consists of a $v_{i,j}^{\tau(x,y_d)+1}$-$v_{i,j}^{m+1,b}$-path of length $m - \tau(x,y_d)$, the edge $\{v_{i,j}^{m+1,b}, b_{i,j}^{0,v}\}$ of length $r - 2m + 2$, a $b_{i,j}^{0,v}$-$b_{i,j}^{\tau(x,y_0)-1}$-path of length $\tau(x,y_d) - d - 1$ and the edge $\{b_{i,j}^{\tau(x,y_0)-1}, \beta_i^x\}$ of length $m + d$. The path $B_{i,j}^{x,\lambda}$ has length

$$\text{dist}(\beta_i^x, b_{i,j}^{\tau(x,y_0)-1}) + \text{dist}(b_{i,j}^{\tau(x,y_0)-1}, b_{i,j}^{0,v}) + \text{dist}(b_{i,j}^{0,v}, v_{i,j}^{m+1,b}) + \text{dist}(v_{i,j}^{m+1,b}, v_{i,j}^{\tau(x,y_d)+1}) =$$
$$m + d + \tau(x,y_d) - d - 1 + r - 2m + 2 + m - \tau(x,y_d) = r + 1.$$

Moreover the following Lemma holds. An illustration (Figure 5) and a proof can be found in the appendix.

▶ **Lemma 10.** *If the gadget $G_{i,j}^\lambda$ represents the pair (x,y), then the hubs of $G_{i,j}^\lambda$ hit the shortest path $A_{i,j}^{(x',y'),\lambda}$ if and only if $(x,y) \neq (x',y')$, and the shortest path $B_{i,j}^{x',\lambda}$ if and only if $x \neq x'$.*

Let us now prove Theorem 6. We show that on G, r-HD has a solution of value $k' = 4Ck(k-1) + \binom{k}{2} + k + 3$ for $r = 2^m$ if and only if H contains a clique if size k.

Proof of Theorem 6. Let $r = 2^m$. Suppose that on the constructed graph G, there is a solution of value k' for r-HD. We observe that every vertex has distance less than $2r$ from the vertex ψ. This means that $B_{2r}(\psi)$ contains the entire graph, and therefore there is a hitting set $\mathcal{H}$ of size $|\mathcal{H}| \leq k'$ for $\mathcal{P}_r$.

We will prove that for any $1 \leq i, j \leq k$ there is some (x,y) such that the hitting set $\mathcal{H}$ contains four hubs $u_{i,j}^{\tau(x,y)}, v_{i,j}^{\tau(x,y)}, a_{i,j}^{\tau(x,y)}, b_{i,j}^{\tau(x,y)}$ from every (copy of the) gadget $G_{i,j}$, and that $\mathcal{H}$ contains one hub $\alpha_{i,j}^{x,y}$ for every $1 \leq i < j \leq k$ and one hub β_i^x for every $1 \leq i \leq k$, such that conditions i and ii are satisfied. This implies that H contains a clique of size k.

Fix now i, j such that $1 \leq i < j \leq k$. We prove that the hitting set $\mathcal{H}$ contains some hub $\alpha_{i,j}^{(x,y)}$. Let $\lambda \in \{1, \ldots, C\}$ and denote the vertices of $\mathcal{H}$ that are contained in $G_{i,j}^\lambda$ by $\mathcal{H}_{i,j}^\lambda$. For the sake of contradiction suppose that there is no (x,y) such that $\alpha_{i,j}^{(x,y)} \in \mathcal{H}$. Lemma 9 states that if $|\mathcal{H}_{i,j}^\lambda| \leq 4$, then $\mathcal{H}_{i,j}^\lambda = \left\{u_{i,j}^{\tau(x,y)}, v_{i,j}^{\tau(x,y)}, a_{i,j}^{\tau(x,y)}, b_{i,j}^{\tau(x,y)}\right\}$ for some (x,y) and therefore $\mathcal{H}$ does not hit the path $A_{i,j}^{(x,y),\lambda}$ according to Lemma 10. Hence, we obtain that for all $\lambda \in \{1, \ldots, C\}$ we have $|\mathcal{H}_{i,j}^\lambda| \geq 5$. Moreover, Lemma 9 states that from any gadget $G_{i',j'}$, $(i', j') \neq (i, j)$ we have to choose at least four hubs. This means however that $|\mathcal{H}| \geq 4Ck(k-1) + C$. If we choose $C = k^2$ we obtain that $4Ck(k-1) + C > k'$, so it cannot be that there is no hub $\alpha_{i,j}^{(x,y)} \in \mathcal{H}$. Analogously we can show that $\mathcal{H}$ contains some hub β_i^x for every $1 \leq i \leq k$, if $C = k^2$. To that end, fix $1 \leq i \leq k$ and suppose that there is no x such that $\beta_i^x \in \mathcal{H}$. Again, it follows from Lemmas 9 and 10 that for all $\lambda \in \{1, \ldots, C\}$ we have $|\mathcal{H}_{i,j}^\lambda| \geq 5$, where $H_{i,j}^\lambda$ denotes the vertices of $\mathcal{H}$ that are contained in $G_{i,j}^\lambda$. As we showed previously, for $C = k^2$ this means that $|\mathcal{H}| > k'$, and it follows that there must be some x such that $\beta_i^x \in \mathcal{H}$.

This means that $\mathcal{H}$ contains at least one hub $\alpha_{i,j}^{(x,y)}$ for every $1 \leq i < j \leq k$, at least one hub β_i^x for every $1 \leq i \leq k$ and at least four hubs from every $G_{i,j}^\lambda$. None of these hubs hits the shortest paths $\psi'' - \psi - \psi''$, $\psi'_\alpha - \psi_\alpha - \psi''_\alpha$, or $\psi'_\beta - \psi_\beta - \psi''_\beta$. To hit these three paths, we need three additional hubs. As $\mathcal{H}$ has size at most $k' = 4Ck(k-1) + \binom{k}{2} + k + 3$, it follows that $\mathcal{H}$ contains precisely four hubs from every $G_{i,j}^\lambda$, so every gadget represents indeed a unique pair (x,y). Moreover, for every $1 \leq i < j \leq k$ there is a unique hub $\alpha_{i,j}^{(x,y)}$ and for every $1 \leq i \leq k$ there is a unique hub β_i^x.

It remains to show that the pairs represented by the individual gadgets fulfill properties i and ii. Consider i, j such that $1 \le i < j \le k$ and let $\lambda, \lambda' \in \{1, \dots, C\}$. Let (x, y) and (x', y') be the pairs represented by $G_{i,j}^{\lambda}$ and $G_{j,i}^{\lambda'}$, respectively. Lemma 10 states that the hubs contained in $G_{i,j}^{\lambda}$ and $G_{j,i}^{\lambda'}$ do not hit the shortest paths $A_{i,j}^{(x,y),\lambda}$ and $A_{j,i}^{(x',y'),\lambda'}$. This means that the two paths must be hit through the hubs $\alpha_{i,j}^{(x,y)}$ and $\alpha_{i,j}^{(x',y')}$. Moreover, both hubs must coincide as $\mathcal{H}$ has size k', i.e. we have $(x, y) = (x', y')$, which implies condition i.

For condition ii, let $1 \le i \le k$, let $1 \le j, j' \le k$, and let $\lambda, \lambda' \in \{1, \dots, C\}$. Denote the pairs represented by $G_{i,j}^{\lambda}$ and $G_{i,j'}^{\lambda'}$ by (x, y) and (x', y'), respectively. It follows from Lemma 10 that the shortest paths $B_{i,j}^{x,\lambda}$ and $B_{i,j'}^{x',\lambda'}$ are not hit through the hubs contained in $G_{i,j}^{\lambda}$ and $G_{i,j'}^{\lambda'}$. This means that the paths must be covered through hubs β_i^x and $\beta_i^{x'}$, and as $|\mathcal{H}| = k'$, this is only possible if $x = x'$, i.e. condition ii is satisfied.

This implies that the graph H indeed contains a clique of size k. To prove the other direction, we refer to the appendix. ◄

5 Approximating Shortest Path Covers

In this section, we show how to approximate r-SPC.

▶ **Theorem 7.** r-*SPC admits a polynomial time $O(\log n)$-approximation algorithm.*

We present an algorithm based on the following ideas. It is well-known that the SET COVER problem is equivalent to HITTING SET by swapping the roles of the elements of the universe and the sets in the given set family. Kuhn et al. [24] study the MINIMUM MEMBERSHIP SET COVER (MMSC) problem, where the aim is to minimize the maximum *membership* of any element of the given universe of the SET COVER instance. Here the *membership* of an element is the number of sets of the solution it is contained in. The MMSC problem finds applications in interference minimization in cellular networks, and Kuhn et al. [24] prove that it admits a polynomial-time $O(\log |U|)$-approximation, where U is the given universe, and they show that this is best possible, unless P=NP. Translated to SPARSE-HS, this means that for an instance where $\mathcal{F} = \mathcal{B}$, an $O(\log |\mathcal{F}|)$-approximation can be computed in polynomial time, and this is also best possible, unless P=NP. We show that r-SPC can be reduced to this version of SPARSE-HS.

We first give a simple observation about the SPARSE-HS problem which will be useful later in our proof. Let $(V, \mathcal{F}, \mathcal{B})$ be a set system and let $B, B' \in \mathcal{B}$ be two sets such that $B \subsetneq B'$. If B' contains at most k elements of the hitting set, then B also contains at most k such elements. Hence we obtain the following.

▶ **Observation 11.** *Let $\mathcal{B}$ be a family containing two sets B, B' such that $B \subsetneq B'$. If there exists a solution to SPARSE-HS for $(V, \mathcal{F}, \mathcal{B} \setminus \{B\})$ of sparseness k, then there exists a solution to SPARSE-HS for $(V, \mathcal{F}, \mathcal{B})$ of sparseness k.*

We reduce the r-SPC problem to the MINIMUM MEMBERSHIP SET COVER (MMSC) problem. Formally, an instance of MMSC consists of a universe U and a family $\mathcal{S}$ of subsets of U, and the goal is to choose a set $\mathcal{S}' \subseteq \mathcal{S}$ such that every element in U belongs to at least one set in $\mathcal{S}'$ and that the maximum membership of any element u with respect to $\mathcal{S}'$ is minimal, where the membership of u is defined as the number of sets in $\mathcal{S}'$ containing u.

Recall that, given a weighted graph $G = (V, E)$ and a radius $r > 0$, the r-SPC problem for G is equivalent to the SPARSE-HS problem on universe V with $\mathcal{F} = \mathcal{P}_r$ and $\mathcal{B} = \{B_{2r}(v) \mid v \in V\}$. Based on Observation 11, we first show that if there exists a ball $B \in \mathcal{B}$ which does not contain any shortest path in $\mathcal{P}_r$ completely, we can safely remove it without affecting the solution.

▶ **Lemma 12.** *Let $B \in \mathcal{B}$ which does not contain any shortest path in $\mathcal{P}_r$ completely, i.e., $S \nsubseteq B$ for every $S \in \mathcal{P}_r$. If there exists a solution to r-SPC for $(V, \mathcal{P}_r, \mathcal{B} \setminus \{B\})$ of sparseness k, then there exists a solution to r-SPC for $(V, \mathcal{P}_r, \mathcal{B})$ of sparseness k.*

Proof. First, if $S \cap B = \emptyset$ for every $S \in \mathcal{P}_r$, then there exists a solution for $(V, \mathcal{P}_r, \mathcal{B})$ not containing any vertices of B, and the claim follows. Now assume there is some path $S_B \in \mathcal{P}_r$ intersecting B in some vertex w. We show that there exists a ball $B' \in \mathcal{B}$ such that $B \subsetneq B'$, and thus the lemma follows from Observation 11.

Let v be the center of the ball $B = B_{2r}(v)$. As $S \nsubseteq B$ for every $S \in \mathcal{P}_r$, $\text{dist}(u, v) \le r$ for every $u \in B$, as otherwise the shortest u-v-path would be contained in the ball B of radius $2r$ with a length in $(r, 2r]$, which would then be in $\mathcal{P}_r$. Hence for any two vertices $u, u' \in B$, $\text{dist}(u, u') \le \text{dist}(u, v) + \text{dist}(v, u') \le 2r$, so we have $B \subseteq B_{2r}(w)$. Moreover, it holds that $S_B \subseteq B_{2r}(w)$ as S_B is the vertex set of a path containing w of length in $(r, 2r]$, and it holds that $S_B \nsubseteq B$, which implies $B \subsetneq B_{2r}(w)$. By definition of $\mathcal{B}$ we have $B_{2r}(w) \in \mathcal{B}$, and thus by Observation 11 the lemma follows.   ◀

Lemma 12 means that we may assume w.l.o.g. that for any $B \in \mathcal{B}$ there is some $S_B \in \mathcal{P}_r$ such that $S_B \subseteq B$. We now give the following observations about the relationship among $\mathcal{P}_r$, $\mathcal{B}$, and a hitting set H of $\mathcal{P}_r$.

▶ **Observation 13.** *Let $S \in \mathcal{P}_r$. As S is the set of vertices of a shortest path π of length $\ell(\pi) \in (r, 2r]$, there exists a ball $B_S \in \mathcal{B}$ of radius $2r$, which completely contains S. This, in turn, implies that $H \cap B_S \ne \emptyset$ and $|H \cap S| \le |H \cap B_S|$.*

▶ **Observation 14.** *Let $B \in \mathcal{B}$. If B contains some shortest path set $S_B \in \mathcal{P}_r$, then we have $H \cap B \ne \emptyset$ and $|H \cap S_B| \le |H \cap B|$.*

By Observations 13 and 14, we get the following.

▶ **Lemma 15.** *There exists a solution to SPARSE-HS for $(V, \mathcal{P}_r, \mathcal{B})$ of sparseness k if and only if there exists a solution to SPARSE-HS for $(V, \mathcal{P}_r \cup \mathcal{B}, \mathcal{B} \cup \mathcal{P}_r)$ of sparseness k.*

Proof. Observe that any solution to SPARSE-HS for $(V, \mathcal{P}_r \cup \mathcal{B}, \mathcal{B} \cup \mathcal{P}_r)$ is also a solution to SPARSE-HS for $(V, \mathcal{P}_r, \mathcal{B})$, as $\mathcal{P}_r \subseteq \mathcal{P}_r \cup \mathcal{B}$ and $\mathcal{B} \subseteq \mathcal{B} \cup \mathcal{P}_r$. We now prove that any solution H to SPARSE-HS for $(V, \mathcal{P}_r, \mathcal{B})$ of sparseness k is also a solution to SPARSE-HS for $(V, \mathcal{P}_r \cup \mathcal{B}, \mathcal{B} \cup \mathcal{P}_r)$ of sparseness k. For this, we need to show that for every $S \in \mathcal{P}_r$, $|H \cap S| \le k$ and for every $B \in \mathcal{B}$, $H \cap B \ne \emptyset$. The former statement follows from Observation 13, while the latter follows from Observation 14 where we assume that B contains some $S_B \in \mathcal{P}_r$ due to Lemma 12.   ◀

We now define an instance of the MINIMUM MEMBERSHIP SET COVER with $U = \mathcal{P}_r \cup \mathcal{B}$ and $\mathcal{S} = \{S_u \mid u \in V\}$, where $S_u = \{S \in U \mid u \in S\}$, and prove the following.

▶ **Lemma 16.** *There exists a solution to SPARSE-HS for $(V, \mathcal{P}_r \cup \mathcal{B}, \mathcal{B} \cup \mathcal{P}_r)$ of sparseness k if and only if there exists a solution to MMSC for $(U, \mathcal{S})$ of value k.*

Proof. We will prove that if there exists a solution to SPARSE-HS for $(V, \mathcal{P}_r \cup \mathcal{B}, \mathcal{B} \cup \mathcal{P}_r)$ of sparseness k, then there exists a solution to MMSC for $(U, \mathcal{S})$ of value k. The proof for the other direction is symmetric. Let H be a solution to SPARSE-HS for $(V, \mathcal{P}_r \cup \mathcal{B}, \mathcal{B} \cup \mathcal{P}_r)$ of sparseness k. We claim that the set $W = \{S_u \in \mathcal{S} \mid u \in H\}$ is a solution to MMSC for $(U, \mathcal{S})$ of value k. Let $E \in U$. Then, $H \cap E \ne \emptyset$. Let $u \in H \cap E$. By the definition of S_u, this implies that $E \in S_u$. Moreover, for any $B \in \mathcal{P}_r \cup \mathcal{B}$, we have that $|H \cap B| \le k$. This implies that B belongs to at most k sets in W. Hence, W is a solution to MMSC for $(U, \mathcal{S})$ of value k.   ◀

Since there exists a $O(\log |U|)$-approximation algorithm for MMSC by Kuhn et al. [24] and $|U| = |\mathcal{P}_r \cup \mathcal{B}| = O(n^2)$, by the above lemma we get an $O(\log n)$-approximation algorithm for r-SPC. This concludes the proof of Theorem 7.

References

1   Ittai Abraham, Daniel Delling, Amos Fiat, Andrew V Goldberg, and Renato F Werneck. Vc-dimension and shortest path algorithms. In *International Colloquium on Automata, Languages, and Programming*, pages 690–699. Springer, 2011.

2   Ittai Abraham, Amos Fiat, Andrew V Goldberg, and Renato F Werneck. Highway dimension, shortest paths, and provably efficient algorithms. In *Proceedings of the twenty-first annual ACM-SIAM symposium on Discrete Algorithms*, pages 782–793. SIAM, 2010.

3   Akanksha Agrawal, Pratibha Choudhary, NS Narayanaswamy, KK Nisha, and Vijayaragunathan Ramamoorthi. Parameterized complexity of minimum membership dominating set. In *International Conference and Workshops on Algorithms and Computation*, pages 288–299. Springer, 2022.

4   Holger Bast, Stefan Funke, Domagoj Matijevic, Peter Sanders, and Dominik Schultes. In transit to constant time shortest-path queries in road networks. In *2007 Proceedings of the Ninth Workshop on Algorithm Engineering and Experiments (ALENEX)*, pages 46–59. SIAM, 2007.

5   Amariah Becker, Philip N Klein, and David Saulpic. Polynomial-time approximation schemes for k-center, k-median, and capacitated vehicle routing in bounded highway dimension. In *26th Annual European Symposium on Algorithms (ESA 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

6   Johannes Blum. Hierarchy of transportation network parameters and hardness results. In *14th International Symposium on Parameterized and Exact Computation (IPEC 2019)*, 2019.

7   Martin Böhm, Ruben Hoeksma, Nicole Megow, Lukas Nölke, and Bertrand Simon. On hop-constrained steiner trees in tree-like metrics. *SIAM Journal on Discrete Mathematics*, 36(2):1249–1273, 2022.

8   Vladimir Braverman, Shaofeng H-C Jiang, Robert Krauthgamer, and Xuan Wu. Coresets for clustering in excluded-minor graphs and beyond. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2679–2696. SIAM, 2021.

9   Karl Bringmann, László Kozma, Shay Moran, and NS Narayanaswamy. Hitting set for hypergraphs of low vc-dimension. In *24th Annual European Symposium on Algorithms (ESA 2016)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2016.

10   Jianer Chen, Iyad A. Kanj, and Ge Xia. Improved parameterized upper bounds for vertex cover. In Rastislav Kralovic and Pawel Urzyczyn, editors, *Mathematical Foundations of Computer Science 2006, 31st International Symposium, MFCS 2006, Stará Lesná, Slovakia, August 28-September 1, 2006, Proceedings*, volume 4162 of *Lecture Notes in Computer Science*, pages 238–249. Springer, 2006. `doi:10.1007/11821069_21`.

11   Yann Disser, Andreas Emil Feldmann, Max Klimm, and Jochen Könemann. Travelling on graphs with small highway dimension. *Algorithmica*, 83(5):1352–1370, 2021. `doi:10.1007/s00453-020-00785-5`.

12   Andreas Emil Feldmann. Fixed-parameter approximations for k-center problems in low highway dimension graphs. *Algorithmica*, 81(3):1031–1052, 2019. `doi:10.1007/s00453-018-0455-0`.

13   Andreas Emil Feldmann, Wai Shing Fung, Jochen Könemann, and Ian Post. A $(1+\varepsilon)$-embedding of low highway dimension graphs into bounded treewidth graphs. *SIAM Journal on Computing*, 47(4):1667–1704, 2018.

14   Andreas Emil Feldmann and David Saulpic. Polynomial time approximation schemes for clustering in low highway dimension graphs. *J. Comput. Syst. Sci.*, 122:72–93, 2021. `doi:10.1016/j.jcss.2021.06.002`.

15   Martin Fürer and Balaji Raghavachari. Approximating the minimum degree spanning tree to within one from the optimal degree. In *Proceedings of the third annual ACM-SIAM symposium on Discrete algorithms*, pages 317–324, 1992.

16   M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-completeness*. W. H. Freeman, 1979.

17   Michael R Garey, David S Johnson, and Larry Stockmeyer. Some simplified np-complete problems. In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 47–63, 1974.

18   Ashwin Jacob, Venkatesh Raman, and Vibha Sahlot. Deconstructing parameterized hardness of fair vertex deletion problems. In *International Computing and Combinatorics Conference*, pages 325–337. Springer, 2019.

19   Aditya Jayaprakash and Mohammad R Salavatipour. Approximation schemes for capacitated vehicle routing on graphs of bounded treewidth, bounded doubling, or highway dimension. *arXiv preprint*, 2021. `arXiv:2106.15034`.

20   Lawqueen Kanesh, Soumen Maity, Komal Muluk, and Saket Saurabh. Parameterized complexity of fair feedback vertex set problem. *Theoretical Computer Science*, 867:1–12, 2021.

21   CS Karthik, Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *Journal of the ACM*, 66(5), 2019.

22   Dusan Knop, Tomás Masarík, and Tomás Toufar. Parameterized complexity of fair vertex evaluation problems. In *44th International Symposium on Mathematical Foundations of Computer Science (MFCS 2019)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019.

23   Petr Kolman, Bernard Lidický, and Jean-Sébastien Sereni. Fair edge deletion problems on tree-decomposable graphs and improper colorings, 2010. Technical report.

24   Fabian Kuhn, Pascal von Rickenbach, Roger Wattenhofer, Emo Welzl, and Aaron Zollinger. Interference in cellular networks: The minimum membership set cover problem. In Lusheng Wang, editor, *Computing and Combinatorics, 11th Annual International Conference, COCOON 2005, Kunming, China, August 16-29, 2005, Proceedings*, volume 3595 of *Lecture Notes in Computer Science*, pages 188–198. Springer, 2005.

25   Lishin Lin and Sartaj Sahni. Fair edge deletion problems. *IEEE transactions on computers*, 38(5):756–761, 1989.

26   Tomáš Masařík and Tomáš Toufar. Parameterized complexity of fair deletion problems. *Discrete Applied Mathematics*, 278:51–61, 2020.

27   J. Maňuch and D. R. Gaur. Fitting protein chains to cubic lattice is NP-complete. *Journal of Bioinformatics and Computational Biology*, 06.01:93–106, 2008.

28   David P Williamson and David B Shmoys. *The design of approximation algorithms*. Cambridge university press, 2011.
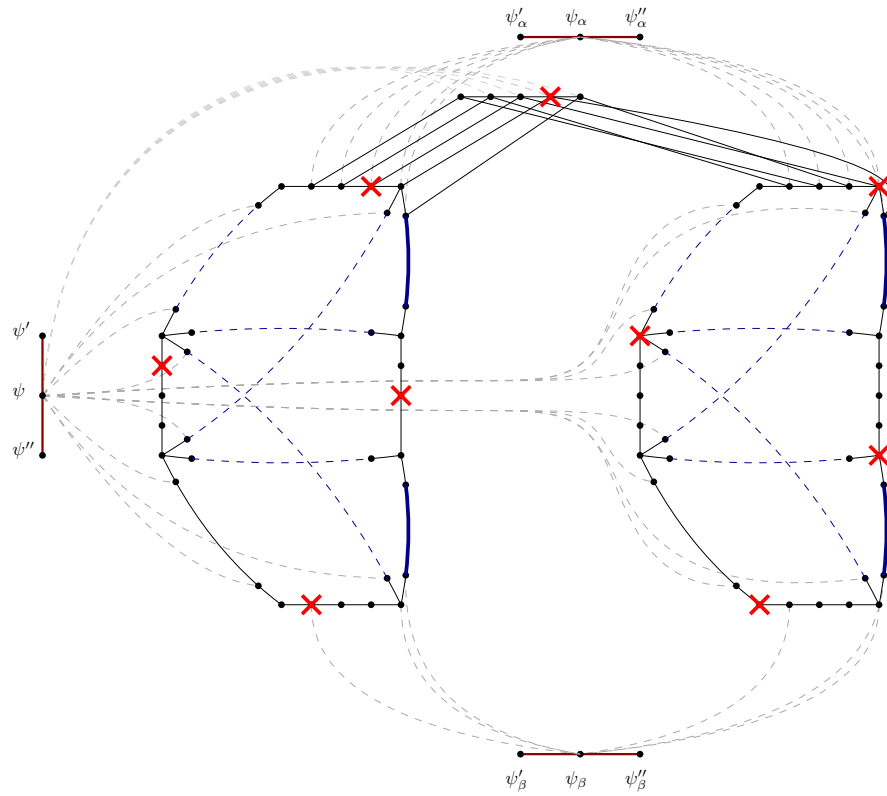
A   Supplementary figures from Section 4

Figure 3 Two gadgets $G_{i,j}$ and $G_{j,i}$ and the connections between them. The marked vertices indicate that $G_{i,j}$ and $G_{j,i}$ represent the pairs (x, y) and (y, x), respectively. Moreover, the vertex $\alpha_{i,j}^{(x,y)}$ is marked.
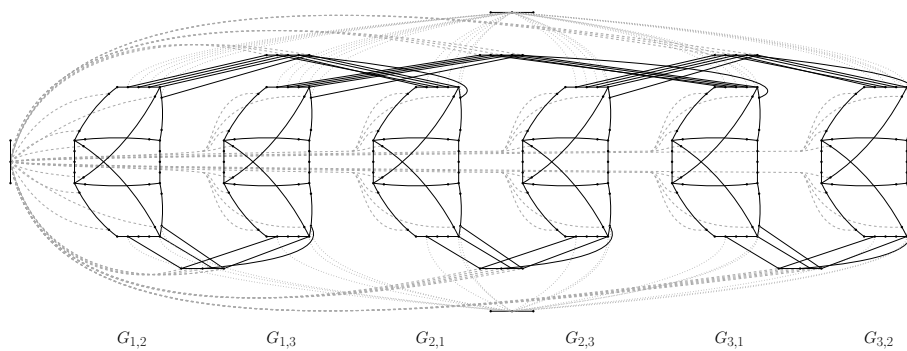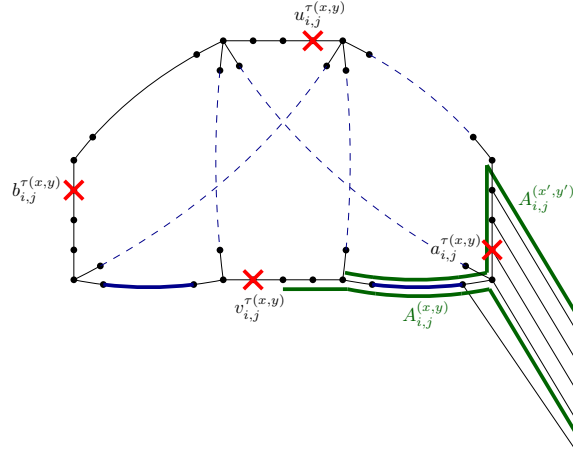

Figure 4 A (simplified) illustration of the whole construction. Note that only one copy $G_{i,j}^{\lambda}$ of every $G_{i,j}$ is shown.

Figure 5 An illustration of Lemma 10. The gadget $G_{i,j}$ represents (x,y), which means that the shortest path $A_{i,j}^{(x,y)}$ is not hit by the hubs of $G_{i,j}$, whereas any other shortest path $A_{i,j}^{(x',y')}$ is hit.

B   Omitted proofs from Section 4

Proof of Lemma 9. Let $1 \le \iota \le m+1$. For $z \in \{a,v,b\}$ we define the path $P^{uz}(\iota)$ as the shortest s-t-path path for

$$s = \begin{cases} u_{i,j}^{\iota} & \text{if } \iota \le m \\ u_{i,j}^{m+1,z} & \text{else} \end{cases} \quad \text{and} \quad t = \begin{cases} z_{i,j}^{\iota-1} & \text{if } \iota > 1 \\ z_{i,j}^{0,u} & \text{else} \end{cases}.$$

Similarly we define the path $P^{zu}(\iota)$ as the shortest s-t-path for

$$s = \begin{cases} z_{i,j}^{\iota} & \text{if } \iota \le m \\ z_{i,j}^{m+1,u} & \text{else} \end{cases} \quad \text{and} \quad t = \begin{cases} u_{i,j}^{\iota-1} & \text{if } \iota > 1 \\ u_{i,j}^{0,z} & \text{else} \end{cases}.$$

We observe that $P^{uz}(\iota)$ passes through the vertices $u_{i,j}^{m+1,z}$ and $z_{i,j}^{0,u}$, and that its length is $(m+1-\iota) + (r-m+1) + (\iota-1) = r+1$. Similarly, $P^{zu}(\iota)$ passes through $z_{i,j}^{m+1,u}$ and $u_{i,j}^{0,z}$, and has also length $r+1$. This means that both $P^{uz}(\iota)$ and $P^{zu}(\iota)$ need to be hit by $\mathcal{H}_{i,j}$. Consider the eight shortest paths $P^{ua}(1), P^{ua}(m+1), P^{au}(1), P^{uv}(m+1), P^{vu}(1)$, $P^{bu}(1), P^{ub}(m+1)$, and $P^{bu}(m+1)$. It holds that every vertex of $G_{i,j}$ covers at most two of these paths, which implies $|\mathcal{H}_{i,j}| \ge 4$. To hit the shortest path $P^{uv}(m+1)$ we have to choose one of the vertices $u_{i,j}^{m+1,v}, v_{i,j}^{0,u}, v_{i,j}^{1}, \dots, v_{i,j}^{m}$. However, the vertices $u_{i,j}^{m+1,v}$ and $v_{i,j}^{0,u}$ do not hit any of the other seven shortest paths. Hence, if we have $|\mathcal{H}_{i,j}| = 4$, then one of the four hubs must be the vertex $v_{i,j}^{\tau(x,y)}$ for some $(x,y) \in E$. Repeating the same argument for the paths $P^{au}(1), P^{ub}(m+1)$, and $P^{ua}(1)$, one can show that if $|\mathcal{H}_{i,j}| = 4$, then $\mathcal{H}_{i,j}$ consists of four vertices $v_{i,j}^{\tau(x,y)}, a_{i,j}^{\tau(x',y')}, v_{i,j}^{\tau(x'',y'')}, b_{i,j}^{\tau(x''',y''')}$.

We now show that for these four vertices it holds that $(x,y) = (x',y') = (x'',y'') = (x''',y''')$. Suppose that for some (x,y) we have $v_{i,j}^{\tau(x,y)} \in \mathcal{H}_{i,j}$. Consider the two shortest paths $P^{uv}(\tau(x,y))$ and $P^{vu}(\tau(x,y))$. Our previous observations imply that both paths must be hit through some vertex $u_{i,j}^{\tau(x',y')}$. As $\mathcal{H}_{ij}$ contains precisely one vertex $u_{i,j}^{\tau(x',y')}$ and as both paths intersect only in $u_{i,j}^{\tau(x,y)}$, it follows that $u_{i,j}^{\tau(x,y)} \in \mathcal{H}_{i,j}$. Similarly, it follows that $\mathcal{H}_{i,j}$ needs to contain the vertices $a_{i,j}^{\tau(x,y)}$ and $b_{i,j}^{\tau(x,y)}$. Hence, if $|\mathcal{H}_{i,j}| = 4$, it follows that there is a unique (x,y) such that $\mathcal{H}_{i,j} = \left\{ u_{i,j}^{\tau(x,y)}, a_{i,j}^{\tau(x,y)}, v_{i,j}^{\tau(x,y)}, b_{i,j}^{\tau(x,y)} \right\}$.   ◀

Proof of Lemma 10. If the gadget $G_{i,j}^\lambda$ represents the pair (x,y), then the hubs of $G_{i,j}^\lambda$ are $u_{i,j}^{\tau(x,y)}, v_{i,j}^{\tau(x,y)}, a_{i,j}^{\tau(x,y)}$, and $b_{i,j}^{\tau(x,y)}$. The shortest path $A_{i,j}^{(x',y'),\lambda}$ contains the vertices $v_{i,j}^{\tau(x',y')-1}, \ldots, v_{i,j}^1$ and the vertices $a_{i,j}^m, \ldots, a_{i,j}^{\tau(x',y')+1}$. This means that $A_{i,j}^{(x',y'),\lambda}$ is hit if and only if $(x,y) \neq (x',y')$. The shortest path $B_{i,j}^{x'}$ contains the vertices $v_{i,j}^{\tau(x',y_d)+1}, \ldots, v_{i,j}^m$ and the vertices $b_{i,j}^1, \ldots, a_{i,j}^{\tau(x',y_0)-1}$, which means that $B_{i,j}^{x'}$ is hit if and only if $x \neq x'$.　◄

Proof of Theorem 6 (continued). For the other direction suppose that the graph H contains a clique $\{w_1, \ldots, w_k\}$ of size k. Consider the following set $\mathcal{H}$: For $1 \leq i,j \leq k, i \neq j$ it contains all C copies of the vertices $u_{i,j}^{\tau(w_i,w_j)}, v_{i,j}^{\tau(w_i,w_j)}, a_{i,j}^{\tau(w_i,w_j)}, b_{i,j}^{\tau(w_i,w_j)}$, for $1 \leq i < j \leq k$ it contains $\alpha_{i,j}^{(w_i,w_j)}$, for $1 \leq i \leq k$ it contains $\beta_{i,j}^{w_i}$, and moreover it contains the three vertices $\psi, \psi_\alpha, \psi_\beta$. It holds that $\mathcal{H}$ has size k'.

We can observe that all shortest paths between different gadgets $G_{i,j}$ and $G_{i',j'}$ are hit by ψ_α or ψ_β. We now show that all shortest paths from $\mathcal{P}_r$ that intersect only one gadget $G_{i,j}$ are hit by $\mathcal{H}$. Let $1 \leq i,j \leq k$ such that $i \neq j$. Suppose that $i < j$, the case $i > j$ can be shown similarly. Consider some vertex t contained in $G_{i,j}$ and denote the shortest path between $\alpha_{i,j}^{(x,y)}$ and t by P. Suppose that P is not hit by ψ. We can observe that the shortest path between $\alpha_{i,j}^{(x,y)}$ and $u_{i,j}^{m+1,a}$ or $b_{i,j}^{0,u}$ contains ψ. As P is not hit by ψ, it follows that

(a) $t = a_{i,j}^\iota$ for some ι or $t \in \{a_{i,j}^{0,u}, a_{i,j}^{m+1,u}, a_{i,j}^{m+1,v}\}$,

(b) $t = v_{i,j}^{\tau(x',y')}$ for some (x',y'), or

(c) $t \in \{v_{i,j}^{0,a}, v_{i,j}^{0,u}, v_{i,j}^{m+1,u}, v_{i,j}^{m+1,b}\}$.

In case a it holds that P has length

$$\operatorname{dist}(\alpha_{i,j}^{(x,y)}, a_{i,j}^{\tau(x,y)+1}) + \operatorname{dist}(a_{i,j}^{\tau(x,y)+1}, t) \leq m + m + 1 = 2m + 1 < r,$$

so it does not need to be hit by $\mathcal{H}$. In case b, the length of P is

$$\operatorname{dist}(\alpha_{i,j}^{(x,y)}, a_{i,j}^{\tau(x,y)+1}) + \operatorname{dist}(a_{i,j}^{\tau(x,y)+1}, v_{i,j}^{0,a}) + \operatorname{dist}(v_{i,j}^{0,a}, v_{i,j}^{\tau(x',y')}) =$$
$$m + m - \tau(x,y) + r - 2m + 2 + \tau(x',y') = r + 2 + \tau(x',y') - \tau(x,y).$$

It follows that the length of P exceeds r if and only if $\tau(x',y') \geq \tau(x,y) - 1$, i.e. the path P contains $A_{i,j}^{(x,y)}$ as a subpath. Lemma 10 states that this subpath is hit by the hubs within $G_{i,j}$ if $(x,y) \neq (w_i, w_j)$, otherwise it is hit by $\alpha_{i,j}^{(x,y)}$. Finally, in case c it holds that P is shorter than r or that P contains $A_{i,j}^{(x,y)}$ as a subpath, which is hit by $\mathcal{H}$, as we just observed.

Analogously, consider some vertex t contained in $G_{i,j}$, denote the shortest path between β_i^x and t by P' and suppose that P' is not hit by ψ. As the shortest path between β_i^x and $u_{i,j}^{0,b}$ or $a_{i,j}^{m+1,u}$ contains ψ, it follows that

(a) $t = b_{i,j}^\iota$ for some ι or $t \in \{b_{i,j}^{0,u}, b_{i,j}^{0,v}, b_{i,j}^{m+1,u}\}$,

(b) $t = v_{i,j}^{\tau(x',y')}$ for some (x',y'), or

(c) $t \in \{v_{i,j}^{0,a}, v_{i,j}^{0,u}, v_{i,j}^{m+1,u}, v_{i,j}^{m+1,b}\}$.

In case a it holds that P' is shorter than r. In case b, the length of P' is

$$\operatorname{dist}(\beta_i^x, b_{i,j}^{\tau(x,y_0)-1}) + \operatorname{dist}(b_{i,j}^{\tau(x,y_0)-1}, v_{i,j}^{m+1,b}) + \operatorname{dist}(v_{i,j}^{m+1,b}, v_{i,j}^{\tau(x',y')}) =$$
$$m + d + \tau(x,y_0) - 1 + r - 2m + 2 + m + 1 - \tau(x',y') =$$
$$r + 2 + \tau(x,y_0) + d - \tau(x',y') = r + 2 + \tau(x,y_d) - \tau(x',y').$$

It holds that the length of P' exceeds r if and only if $\tau(x',y') \leq \tau(x,y_d)+1$, which is the case if and only if P' contains $B_{i,j}^x$ as a subpath, which is hit by the hubs within $G_{i,j}$ or by $\beta_i^{(w_i,w_j)}$. In case c it holds that the length of P' is at most r or that P' contains $B_{i,j}^x$ as a subpath, which is hit by $\mathcal{H}$.

Consider now the shortest path between $u_{i,j}^{\tau(x,y)}$ and some vertex t of $G_{i,j}$ and denote it by P''. Define the shortest paths $P^{uz}(\iota)$ and $P^{zu}(\iota)$ as in the proof of Lemma 9. If the length of P' exceeds r then for some $z \in \{a,v,b\}$, the path P'' contains the path $P^{uz}(\tau(x,y))$ to $z_{i,j}^{\tau(x,y)-1}$ or the path $P^{zu}(\tau(x,y)+1)$ to $z_{i,j}^{\tau(x,y)+1}$ as a subpath. Suppose that P'' contains $P^{uz}(\tau(x,y))$, the other case is analogous. To show that $P^{uz}(\tau(x,y))$ (and hence also P'') is hit by $\mathcal{H}$, we distinguish two cases: If $(w_i,w_j) \prec (x,y)$, then $P^{uz}(\tau(x,y))$ is hit through $z_{i,j}^{\tau(w_i,w_j)-1}$, otherwise it is hit through $u_{i,j}^{\tau(w_i,w_j)-1}$. Similarly it can be shown that any shortest path between two vertices of $G_{i,j}$ whose length exceeds r is hit by $\mathcal{H}$, which means that $\mathcal{H}$ is a solution for r-HD on G of value k'.   ◀

C   Dense Matching

▶ **Theorem 8.** *It is NP-hard to approximate* DENSE MATCHING *within* $2-\varepsilon$ *for any* $\varepsilon > 0$, *even if* $\mathcal{B} = \{B_2(v) \mid v \in V\}$ *where all edges have weight* 1.

Proof. Consider the following reduction from 3-SAT. Let an instance of this problem be given by a set of variables $X = \{x_i\}_{i=1,\dots,n}$ and a set of clauses $\mathcal{C} = \{C_j\}_{j=1,\dots,m}$ with $C_j \subset X \cup \bar{X}, |C_j| \leq 3$. Let $\bar{\bar{x}} = x$. We construct the graph $G = (V,E)$ given by

$$V = \bigcup_{i=1}^n \left(\{x_i,\bar{x}_i,x^0\} \cup \{x_i^\ell,\bar{x}_i^\ell \mid 1 \leq \ell \leq 7\}\right) \cup \bigcup_{j=1}^m \left(\{z_j\} \cup \bigcup_{x \in C_j} \{x^{j,\ell} \mid 1 \leq \ell \leq 4\}\right)$$

and

$$E = \bigcup_{i=1}^n \left(\{\{x_i,x_i^0\},\{x_i,x_i^1\},\{x_i^1,x_i^0\},\{\bar{x}_i,x_i^0\},\{\bar{x}_i,\bar{x}_i^1\},\{\bar{x}_i^1,x_i^0\}\}\right) \cup$$

$$\bigcup_{i=1}^n \left(\bigcup_{\ell=1}^6 \{\{x_i^\ell,x_i^{\ell+1}\},\{\bar{x}_i^\ell,\bar{x}_i^{\ell+1}\}\} \cup \{\{x_i^7,x_i^4\},\{\bar{x}_i^7,\bar{x}_i^4\}\}\right) \cup$$

$$\bigcup_{j=1}^m \bigcup_{x \in C_j} \{\{z_j,x^{j,1}\},\{x^{j,1},x^{j,2}\},\{x^{j,2},x^{j,3}\},\{x^{j,3},x^{j,4}\},\{x^{j,4},x\},\{x^{j,4},x^0\}\}$$

The construction is illustrated in Figure 6.

We now show that the given 3-SAT formula is satisfiable if and only if there is a matching M such that $|M \cap E(B_2(v))| \geq 2$ for every ball $B_2(v)$ of radius 2, where we assume that edges have unit length. This means that if the given formula is not satisfiable, then there is a ball $B_2(v)$ such that $|M \cap E(B_2(v))| \leq 1$, which implies that it is NP-hard to obtain an approximation factor less than two.

Suppose that the given formula has a satisfying assignment $\alpha \colon X \to \{0,1\}$ and extend α to $\bar{X}$ by choosing $\alpha(\bar{x}) = 1 - \alpha(x)$. For $j = 1 \dots m$ let $y_j \in C_j$ be some literal satisfying C_j, i.e. $\alpha(y_j) = 1$. We construct the matching

$$M = \bigcup_{i=1}^n \{\{x_i^2,x_i^3\},\{x_i^4,x_i^5\},\{x_i^6,x_i^7\},\{\bar{x}_i^2,\bar{x}_i^3\},\{\bar{x}_i^4,\bar{x}_i^5\},\{\bar{x}_i^6,\bar{x}_i^7\},\} \cup \bigcup_{x \colon \alpha(x)=1} \{\{x,x^0\}\}$$

$$\cup \bigcup_{x \colon \alpha(x)=0} \{\{x,x^1\}\} \cup \bigcup_{j=1}^m \left(\{\{z_j,y_j^{j,1}\},\{y_j^{j,2},y_j^{j,3}\}\} \cup \bigcup_{x \in C_j \setminus \{y_j\}} \{\{x^{j,1},x^{j,2}\},\{x^{j,3},x^{j,4}\}\}\right)$$
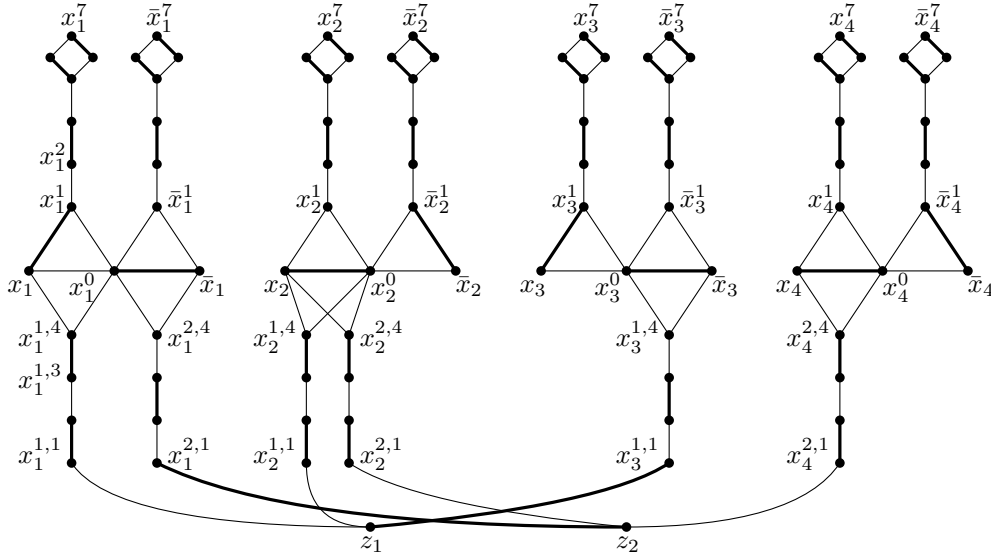
Figure 6 The graph G for the formula $(x_1 \vee x_2 \vee \bar{x}_3) \wedge (\bar{x}_1 \vee x_2 \vee x_4)$. The bold edges yield a matching that corresponds to the assignment $x_1 \mapsto 0, x_2 \mapsto 1, x_3 \mapsto 0, x_4 \mapsto 1$.

It is easy to verify that M is indeed a matching and that $|M \cap E(B_2(v))| \geq 2$ for every ball $B_2(v)$.

Suppose now that there is some matching M such that $|M \cap E(B_2(v))| \geq 2$ for every ball $B_2(v)$. Consider the assignment $\alpha \colon X \to \{0, 1\}, \alpha(x) = 1$ if and only if there is some j such that $\{z_j, x^{j,1}\} \in M$. To show that α is a satisfying assignment, consider some clause C_j and let $C_j = \{x_{i_1}, x_{i_2}, x_{i_3}\}$. Consider the ball $B_2(x_{i_1}^{j,1}) = \{z_j, x_{i_1}^{j,1}, x_{i_1}^{j,2}, x_{i_1}^{j,3}, x_{i_2}^{j,1}, x_{i_3}^{j,1}\}$. We show that M contains one of the edges $\{z_j, x_{i_1}^{j,1}\}, \{z_j, x_{i_2}^{j,1}\}$, and $\{z_j, x_{i_3}^{j,1}\}$. To prove this, suppose that M contains none of these three edges. This means that M has to contain the two remaining edges $\{x_{i_1}^{j,1}, x_{i_1}^{j,2}\}$ and $\{x_{i_1}^{j,2}, x_{i_1}^{j,3}\}$ contained in $E(B_2(x_{i_1}^{j,1}))$, which is not possible as both edges are incident to $x_{i_1}^{j,2}$.

Let now $\{z_j, x_\iota^{j,1}\}$ be the edge contained in M. If $x_\iota \in X$, i.e. x_ι is a positive literal, it immediately follows that α satisfies the clause C_j. Suppose now that $x_\iota \in \bar{X}$. We show that in this case we have $\alpha(x_\iota) = 0$, i.e. there is no j' such that $\{z_{j'}, \bar{x}_\iota^{j',1}\} \in M$. For the sake of contradiction, suppose that $\{z_{j'}, \bar{x}_\iota^{j',1}\} \in M$ for some $j' \in \{1, \ldots, m\}$. It follows from $|M \cap E(B_2(x_\iota^{j,1}))| \geq 2$ that $\{x_\iota^{j,2}, x_\iota^{j,3}\} \in M$. Consider the ball $B_2(x_\iota^{j,3}) = \{x_\iota^{j,1}, x_\iota^{j,2}, x_\iota^{j,3}, x_\iota^{j,4}, x_\iota, \bar{x}_\iota^0\}$. As M contains two edges from this ball and one of these edges is $\{x_\iota^{j,2}, x_\iota^{j,3}\}$, the other edge needs to be contained in the triangle $\{x_\iota^{j,4}, x_\iota, \bar{x}_\iota^0\}$. Consider now the ball $B_2(x_\iota^7) = \{x_\iota^4, x_\iota^5, x_\iota^6, x_\iota^7\}$. It holds that $\{x_\iota^4, x_\iota^5\} \in M$ or $\{x_\iota^4, x_\iota^6\} \in M$, which implies $\{x_\iota^3, x_\iota^4\} \notin M$. If we now consider the ball $B_2(x_\iota^2) = \{x_\iota, \bar{x}_\iota^0, x_\iota^1, x_\iota^2, x_\iota^3, x_\iota^4\}$, it follows that M needs to contain one edge from the triangle $\{x_\iota, \bar{x}_\iota^0, x_\iota^1\}$. As M also contains one edge from the triangle $\{x_\iota^{j,4}, x_\iota, \bar{x}_\iota^0\}$, we obtain that M contains $\{\bar{x}_\iota^0, x_\iota^4\}, \{\bar{x}_\iota^0, x_\iota\}$, or $\{\bar{x}_\iota^0, x_\iota^1\}$.

However, as we also have $\{z_{j'}, \bar{x}_\iota^{j',1}\} \in M$, it follows analogously that M contains $\{\bar{x}_\iota^0, \bar{x}_\iota^4\}, \{\bar{x}_\iota^0, \bar{x}\}$, or $\{\bar{x}_\iota^0, \bar{x}_\iota^1\}$, which is not possible. This means that α is indeed a satisfying assignment, which concludes the proof.   ◀