Some Results on Approximability of Minimum Sum Vertex Cover

Aleksa Stanković

Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden

Abstract

We study the Minimum Sum Vertex Cover problem, which asks for an ordering of vertices in a graph that minimizes the total cover time of edges. In particular, n vertices of the graph are visited according to an ordering, and for each edge this induces the first time it is covered. The goal of the problem is to find the ordering which minimizes the sum of the cover times over all edges in the graph.

In this work we give the first explicit hardness of approximation result for Minimum Sum Vertex Cover. In particular, assuming the Unique Games Conjecture, we show that the Minimum Sum Vertex Cover problem cannot be approximated within 1.014. The best approximation ratio for Minimum Sum Vertex Cover as of now is 16/9, due to a recent work of Bansal, Batra, Farhadi, and Tetali.

We also revisit an approximation algorithm for regular graphs outlined in the work of Feige, Lovász, and Tetali, and show that Minimum Sum Vertex Cover can be approximated within 1.225 on regular graphs.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness; Mathematics of computing → Approximation algorithms; Theory of computation → Approximation algorithms analysis

Keywords and phrases Hardness of approximation, approximability, approximation algorithms, Label Cover, Unique Games Conjecture, Vertex Cover

Digital Object Identifier 10.4230/LIPIcs.APPROX/RANDOM.2022.50

Category APPROX

Funding Research supported by the Approximability and Proof Complexity project funded by the Knut and Alice Wallenberg Foundation.

Acknowledgements Author thanks Johan Håstad for fruitful discussion, as well as Hans Oude Groeniger and anonymous reviewers for useful comments which improved the presentation of this work.

1 Introduction

In the Minimum Sum Vertex Cover problem, as an input we are given a graph $G = (V, E)$, and the goal is to find an ordering of vertices which minimizes the total cover time of edges in E. In particular, we visit vertices in $|V|$ steps, one at each step, and an edge e is considered to be covered at the time $t \in \{1, \ldots, |V|\}$ if the first time one of its endpoints is visited by the ordering is t.

The Minimum Sum Vertex Cover (MSVC) problem was introduced by Feige, Lovász, and Tetali [9], as a special case of the Minimum Sum Set Cover problem, which was of primary interest in that work. The same work showed that MSVC can be approximated within a factor of 2 using linear programming. That work also studied MSVC on regular graphs, and observed that a greedy algorithm approximates the optimal value within a factor of 4/3. In addition to this, it was shown that 4/3 factor can be improved using semidefinite programming to some non-explicit constant β smaller than 4/3.

The 2-approximation algorithm was subsequently improved by Barenholz, Feige, and Peleg [7], who gave a 1.999946-approximation algorithm for this problem. This was then substantially improved by Bansal, Batra, Farhadi, and Tetali, who, using linear programming with fairly involved rounding procedure, showed that MSVC can be approximated within a factor of 16/9. Furthermore, the same work gives a linear programming integrality gap matching the approximation ratio.

So far explicit hardness of approximation results for this problem have been lacking, and to the best knowledge of the author, the only inapproximability result [9] gives hardness of $1+\varepsilon$, for some small non-explicit $\varepsilon > 0$, using a reduction from the Minimum Vertex Cover problem on bounded degree graphs [1, 4]. In this work we give the first explicit hardness for MSVC, which we state in the following theorem.

▶ **Theorem 1.** *Assuming the Unique Games Conjecture, Minimum Sum Vertex Cover is NP-hard to approximate within* 1.014*.*

We use the Unique Games Conjecture introduced by Khot [10] as our hardness assumption. This conjecture has been the central open problem in the hardness of approximation area since its introduction, and many already known (and optimal) hardness of approximation results rely on the validity of this conjecture [15, 2, 13, 6].

Furthermore, our hardness reduction outputs regular graphs, for which better approximation algorithms are known compared to the general case.

Further to this, we will also revisit the approximation algorithm of Feige, Lovász, and Tetali [9] for regular graphs. Our contribution can be described as follows. The algorithm for regular graphs outlined in [9] uses an approximation algorithm for a problem called Max-k-VC in a "black box" manner. Max-k-VC problem is the problem of finding k vertices in a graph that cover as many edges as possible. The approximation ratio of the algorithm for regular graphs in [9] depends on the approximation ratio α for Max-k-VC problem. Due to the developments since the publication of [9] on Max-k-VC, a better value of α can be achieved, and hence by using this value we can obtain stronger approximation. Furthermore, a certain bound[1] used in an argument outlined in [9] for the approximation algorithm on regular graphs is incorrect, which we show by giving a counterexample in the appendix. We correct this by proving the optimal bound, and observe that the rest of the argument still holds. Let us remark that the sharpness of the bound affects the approximation ratio, and hence finding the optimal bound is desirable in this case. In conclusion, we obtain the following result

▶ **Theorem 2.** *Minimum Sum Vertex Cover can be approximated within* 1.225 *on regular graphs.*

1.1 Techniques and Proof Ideas

In this section we give an overview of the proof and briefly discuss techniques used.

The starting point of our reduction are Unique Games, which we formally describe in Section 2. More precisely, we use regular Affine Unique Games as an input to our reduction. Regular Affine Unique Games are Unique Games in which the alphabet is understood as an additive group $\mathbb{Z}_L$, and the constraints are of form $x_u - x_v = c_e$ for an edge $e = (u, v)$, while the word regular indicates that the constraint graph is regular. Interestingly, in this

[1] We do not discuss what this bound exactly is here, for the sake of clarity.

work the structure of Affine Unique Games actually helps us achieve better completeness and therefore a stronger inapproximability result. The property of Affine Unique Games that we use can be described as follows. Let us consider the completeness case, in which we have some assignment z of labels to the vertices in the Affine Unique Games, which satisfies almost all the constraints. Then, for any $a \in \mathbb{Z}_L$, the assignment $z_a = z + a$ gives another assignment which satisfies almost all the constraints. Furthermore, if we let $V_a, a \in \mathbb{Z}_L$, to be the vertex subset in the label extended graph comprised of vertex labels "selected" by the map z_a, then the sets V_a are disjoint, and this gives us enough structure to find an ordering with a low sum set cover value.

Let us elaborate. Our reduction uses the same standard long code dictatorship testing as the celebrated paper of Khot, Kindler, Mossel, and O'Donnell [11], which among other results gave the optimal hardness of Max-Cut assuming the Unique Games Conjecture. This is the same reduction that appeared in [4, 5], and hence the graphs that are output by the reduction satisfy the same properties as outlined in these works, which turns out to be useful for studying soundness. In particular, in the soundness case, for each $r \in (0, 1)$, and each vertex subset of fractional size r, we have a lower bound $b := b(r)$ on the number of edges with both endpoints in this subset. Therefore, no matter which order of visiting the vertices we choose, after $t \in \{1, \dots, n\}$ steps, we have not covered edges which have both endpoints in vertices visited after the time t, and hence at the time t we have at least $b(1 - t/n)$ uncovered edges. This gives us a lower bound of form $\sum_{i=1}^{n} b(1 - r/n) \approx \int_0^1 b(x)dx$.

In the completeness case, we are supposed to specify an ordering of the vertices in each of $k \in \mathbb{N}$ long codes. Given this ordering, in the first pass we would pick first vertex in each of the k long codes, after which we would pick the second vertex in each long code, etc. The order in which we visit k long codes will not be impactful. Hence, it is very important to pick order of visiting vertices in each long code well. This is where the affine structure of Unique Games proves to be useful. In the case we have only one good labeling (as it is the case with "classical" Unique Games), an obvious observation is that we can take first all vertices with 0 in the coordinate fixed by a good labelling $z = z_0$, and then all vertices with 1 in the same coordinate. However, there are many vertices in a long code which have 0 in the coordinate fixed by a good labelling, and hence many orderings can be chosen. Therefore, the question is which order should one pick the vertices with in this subset? Since in Affine Unique Games we have a second satisfying assignment, namely z_1, there is a natural ordering among these. We iterate through vertices that have 0 in the coordinate fixed by z_1, and after visiting the whole subgraph, visit vertices that have 1 in the coordinate fixed by z_1. We can repeat this idea and visit smaller and smaller subgraphs, the last of which will consist only of two vertices and for which we will use z_{L-1}.

The idea of using multiple good assignments in reductions from Unique Games already appeared in [8], but it is still fairly uncommon. Hence, it would be interesting to see whether it would be useful for some other problems as well.

As we mentioned in the introduction, the output of the hardness reduction is a weighted graph, and we need to remove its weights. The idea for this is simple: we replace each vertex v with m new vertices which we group in a set A_v, for m is sufficiently large. We then replace each edge $e = (u, v)$ by sampling edges between A_u and A_v at a correct density. This graph indeed looks like the initial graph and is almost regular, however, proving that it preserves soundness and completeness properties, and making it exactly regular, requires some effort. Due to the size limitation, we defer the details of this part to the full version of this paper.

1.2 Organization

In Section 2 we introduce the notation used in this work, recall some well known facts, and formally introduce the Minimum Sum Vertex Cover problem. Then, in Section 3, we give our hardness reduction which outputs weighted graphs and discuss how it can be used to show Theorem 1.

Then, in Section 4 we show how Minimum Sum Vertex Cover on regular graphs can by approximated within a factor of 1.225, by recalling the algorithm from [9] and making necessary changes.

2 Preliminaries

For $n \in \mathbb{N}$ we use $[n]$ to denote $[n] = \{1, 2, \dots, n\}$. In this paper we work with undirected (multi)graphs $G = (V, E)$. For a set $S \subseteq V$ of vertices we use S^c to denote its complement $S^c = V \setminus S$, and write $U \sqcup V$ for a disjoint union of sets U and V.

The initial graph output by our reduction will be edge weighted. The weights of edges are given by a function $w\colon E \to [0, 1]$. For a subset $K \subseteq E$ we interpret $w(K)$ as the sum of weights of edges in K. Furthermore, we will typically *normalize* the weights so that $w(E) = 1$. For $S, T \subseteq V$, we write $w(S, T)$ for the total weight of edges from E which have one endpoint in S, and other in T. Note that, since we work with undirected graphs, the order of endpoints is not important, and therefore $w(S, T) = w(T, S)$. We remark that the sets S, T, do not need to be disjoint. We also use $N(S, T)$ to denote the set of all edges with one endpoint in S and other endpoint in T. For a vertex $v \in V$, we use $N(v)$ to denote the set of its neighbours.

The following definition will be useful in discussing properties of our reduction.

▶ **Definition 3.** *A graph G is (r, h)-dense if every subset $S \subseteq V$ with $w(S) = r$ satisfies $w(S, S) \geq h$.*

Minimum Sum Vertex Cover is arguably more natural in an unweighted setting, i.e., the setting in which the weights of all edges are equal. Let us now introduce the Min Sum Vertex Cover problem for unweighted graphs.

▶ **Definition 4.** *Consider an unweighted graph $G = (V, E)$, and let $n = |V|$. For an ordering of vertices represented as a bijection $\sigma\colon [n] \leftrightarrow V$, and an edge $(u, v) = e \in E$, let us denote with $c_{\sigma,e}$ the "time" at which edge e is covered, that is*

$$c_{\sigma,e} = \min(\sigma^{-1}(u), \sigma^{-1}(v)).$$

Then the Sum Vertex Cover under scheduling σ, which we denote by $\mathrm{SVC}_G(\sigma)$, is given as

$$\mathrm{SVC}_G(\sigma) = \frac{1}{|E|} \sum_{e \in E} c_{\sigma,e}. \tag{1}$$

The value of Min Sum Vertex Cover is the minimal value of $SVC_G(\sigma)$ over all possible permutations σ, that is

$$\mathrm{MSVC}(G) = \min_{\sigma\colon [n] \leftrightarrow V} \mathrm{SVC}_G(\sigma). \tag{2}$$

We have normalized the expression for $\mathrm{SVC}_G(\sigma)$ by $\frac{1}{|E|}$ for the sake of writing convenience. This does not affect our results, since the normalization factor will be cancelled out when studying approximation ratios. We can also reformulate the expression (1), stating the value of Sum Vertex Cover under scheduling σ, as follows. At the time $t \in [n]$, the total number of edges not covered[2] is $w(\sigma([t])^c, \sigma([t])^c)$, and let us assign them the cost of 1 at that time. The cost $c_{\sigma,e}$ of an edge e under σ is exactly the number of times t the edge was not covered, and hence we can write

$$\mathrm{SVC}_G(\sigma) = \frac{1}{|E|}\sum_{t=1}^{n} w(\sigma([t])^c, \sigma([t])^c). \tag{3}$$

We remark that this allows us to define Mininimum Sum Vertex Cover for edge weighted graphs, by replacing $\frac{1}{|E|}$ above with $\frac{1}{w(E)}$, i.e., for weighted graphs we have

$$\mathrm{SVC}_G(\sigma) = \frac{1}{w(E)}\sum_{t=1}^{n} w(\sigma([t])^c, \sigma([t])^c).$$

We can also discuss Minimum Sum Vertex Cover for weighted graphs in the sense of definitions (1) and (2) by letting

$$\mathrm{SVC}_G(\sigma) = \frac{1}{w(E)}\sum_{e\in E} w_e c_{\sigma,e}.$$

As mentioned in the introduction, we can extend this definition in a natural way to include vertex weights. However, we have not found vertex weights to be useful for hardness reduction, and hence we omit further discussing this for the sake of simplicity.

In order to state the quantities appearing in our result, it is necessary to introduce some more notation. We use $\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$ to denote the density function of a standard normal random variable, and $\Phi(x) = \int_{-\infty}^{x}\phi(y)dy$ to denote its cumulative distribution function (CDF). We also work with bivariate normal random variables, and to that end introduce the following function.

▶ **Definition 5.** *Let $\rho \in [-1,1]$, and consider two jointly normal random variables X, Y, with mean 0, and covariance matrix* $\mathrm{Cov}(X,Y) = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}$. *We define $\Gamma_\rho \colon [0,1]^2 \to [0,1]$ as*

$$\Gamma_\rho(x,y) = \Pr\left[X \le \Phi^{-1}(x) \wedge Y \le \Phi^{-1}(y)\right].$$

We also write $\Gamma_\rho(x) = \Gamma_\rho(x,x)$.

The hardness result stated in this paper is based on the Unique Games Conjecture. In order to state this conjecture, we first introduce Unique Games.

▶ **Definition 6.** *A* Unique Games instance $\Lambda = (\mathcal{U}, \mathcal{V}, \mathcal{E}, \Pi, [L])$ *consists of an unweighted bipartite multigraph $(\mathcal{U} \sqcup \mathcal{V}, \mathcal{E})$, a set $\Pi = \{\pi_e \colon [L] \to [L] \mid e \in \mathcal{E}$ and π_e is a bijection$\}$ of permutation constraints, and a set $[L]$ of labels. The value of Λ under the assignment $z \colon \mathcal{U} \sqcup \mathcal{V} \to [L]$ is the fraction of edges satisfied, where an edge $e = (u,v), u \in \mathcal{U}, v \in \mathcal{V}$, is satisfied if $\pi_e(z(u)) = z(v)$. We write* $\mathrm{Val}_z(\Lambda)$ *for the value of Λ under z, and* $\mathrm{Opt}(\Lambda)$ *for the maximum possible value over all assignments z.*

Let us remark that we require Unique Games instance graph $(\mathcal{U}, \mathcal{V}, \mathcal{E})$ to be regular. Since Unique Games belong to the class of problems known as Constraint Satisfaction Problems (CSPs), without loss of generality we can assume regularity, as shown in [16].

The Unique Games Conjecture [10] can be stated as follows ([12], Lemma 3.4).

[2] We interpret $\sigma([t])$ as $\sigma([t]) = \{\sigma(i) \mid i \in [t]\}$.

▶ **Conjecture 7** (Unique Games Conjecture). *For every constant $\gamma > 0$ there is a sufficiently large $L \in \mathbb{N}$, such that for a Unique Games instance $\Lambda = (\mathcal{U}, \mathcal{V}, \mathcal{E}, \Pi, [L])$ with a regular bipartite graph $(\mathcal{U} \sqcup \mathcal{V}, \mathcal{E})$, it is NP-hard to distinguish between*
- $\mathrm{Opt}(\Lambda) \geq 1 - \gamma$,
- $\mathrm{Opt}(\Lambda) \leq \gamma$.

The starting point of hardness result in this work are Affine Unique Games, which are a type of Unique Games defined as follows.

▶ **Definition 8.** *An Affine* Unique Games instance $\Lambda = (\mathcal{U}, \mathcal{V}, \mathcal{E}, \Pi, [L])$ *is a Unique Games Instance Λ in which all permutation constraints π_e are affine constraints. Furthermore, the alphabet $[L]$ is identified with an additive group $\mathbb{Z}_L$, and for each $\mathcal{E} \ni e = (u, v)$ we have $\pi_e(x) = x - c_e$, where $c_e \in \mathbb{Z}_L$ is a constant.*

We remark that approximating Affine Unique Games is equally hard as approximating Unique Games, in the sense stated by the lemma below which was proved in [11].

▶ **Lemma 9** (Affine Unique Games Hardness). *Assuming the Unique Games Conjecture, the following statement holds. For every constant $\gamma > 0$, there is a sufficiently large $L \in \mathbb{N}$, such that for an Affine Unique Games instance $\Lambda = (\mathcal{U}, \mathcal{V}, \mathcal{E}, \Pi, [L])$ with a regular bipartite graph $(\mathcal{U} \sqcup \mathcal{V}, \mathcal{E})$, it is NP-hard to distinguish between*
- $\mathrm{Opt}(\Lambda) \geq 1 - \gamma$,
- $\mathrm{Opt}(\Lambda) \leq \gamma$.

3 Hardness Reduction

In this section we state and prove our main result. In Section 3.1 we give a reduction from Affine Unique Games to weighted graphs which satisfy properties sufficient for showing hardness of approximating Min Sum Vertex Cover.

3.1 Reduction from Unique Games to Weighted Graphs

We remark that we use the same type of reduction as in [11, 4, 5], with the only difference being that we now use Affine Unique Games as the starting point, and compared to [5] we are here interested only[3] in the unbiased setting ($q = 1/2$). The main challenge lies in proving completeness, since we will reuse the soundness property of the reduction in the aforementioned results.

Before giving the full proof of the result, we will sketch the ideas behind studying the completeness case now. Consider having a labelling z which satisfies almost all the edges. Let us describe what happens locally on two vertices u, v, with a common neighbour w, which are chosen such that (u, w) and (v, w) edges are satisfied by z. For the sake of simplicity, let us assume that the affine constraints on $e_1 = (u, w)$, and $e_2 = (v, w)$, are trivial, that is, $c_{e_1} = c_{e_2} = 0$, so that the labels x_u and x_v are matched by z if and only if $x_u = x_v$. Then, we replace both u and v with 2^L strings of length L. Let us call the sets of strings which replaced u and v as R and S, respectively. We drop indices u, v here for the sake of readability. Hence, we have

$$S = \{(s_1, \ldots, s_L) \mid s_i \in \{0, 1\}, i \in [L]\}, \quad R = \{(r_1, \ldots, r_L) \mid r_i \in \{0, 1\}, i \in [L]\}.$$

[3] We remark that one could also consider using a reduction with biased bits, i.e., the reduction from [5] with $q \neq 1/2$. However, this does not yield better inapproximability.

Edges between S and R are created as follows. The reduction first fixes some negative correlation parameter $\rho \in (-1, 0)$, samples L times pairs of unbiased, ρ correlated bits, $(s_i, r_i), i = 1, \ldots, L$, and then adds an edge between $s = (s_1, \ldots, s_L) \in S$ and $r = (r_1, \ldots, r_L) \in R$. Let us use ν to denote the probability distribution of two ρ correlated, unbiased bits, i.e.,

$$\nu(0,0) = \nu(1,1) = \frac{1+\rho}{4}, \quad \nu(0,1) = \nu(1,0) = \frac{1-\rho}{4},$$

and study Minimum Sum Vertex Cover on this graph. We will upper bound the value of MSVC on this graph G_L by[4] some T_L, by exhibiting an ordering σ_L. Actually, we build our ordering for vertices in G_L by using the ordering on G_{L-1}, which is a graph that would have been created with an alphabet size $L-1$. In particular, we observe that the induced subgraph of G_L obtained by fixing $s_1 = r_1 = 0$ is isomorphic to G_{L-1}. Hence, if we use σ_{L-1} to visit vertices in this subgraph, edges with both endpoints in it will be visited by the time T_{L-1} on average. Since the total weight of edges in this subgraph is $\nu(0,0)$, the cost of covering edges in this subgraph is at most

$$\nu(0,0) \cdot T_{L-1}.$$

We have spent 2^L steps in visiting this subgraph. Observe that we also covered the edges between strings s, r, which have $(s_1, r_1) \in \{(0,1), (1,0)\}$. In particular, we will show that they are covered by the time $2^L/2$ on average, which intuitively can be seen by observing that we visit G_{L-1} in 2^L steps, and an average edge will be visited in half that time. This gives us a cost

$$(\nu(0,1) + \nu(1,0)) \cdot 2^{L-1}.$$

Finally, the subgraph with s, r such that $s_1 = r_1 = 1$ is also isomorphic to G_{L-1}, and once again use the ordering σ_{L-1} to traverse it in 2^L steps. In this case, we have a delay of 2^L due to visiting vertices with $r_1 = 0$ or $s_1 = 0$, and hence the edges are covered by the time $2^L + T_{L-1}$, and their total cost is

$$\nu(1,1) \cdot (2^L + T_{L-1}).$$

Hence, we have that

$$T_L \leq \nu(0,0) \cdot T_{L-1} + (\nu(0,1) + \nu(1,0)) \cdot 2^{L-1} + \nu(1,1) \cdot (2^L + T_{L-1}).$$

Letting $t_L = T_L/2^{L+1}$ and replacing the values of ν yields

$$t_L \leq \frac{1+\rho}{4} t_{L-1} + \frac{1}{4},$$

which is a recurrence relation, and solving it shows that $t_L \to \frac{1}{3-\rho}$, regardless of t_1. Hence, for sufficiently large L we should expect to get MSVC close to $\frac{1}{3-\rho}$.

With this intuition in mind, we now state and prove the theorem which gives the hardness reduction from Affine Unique Games to weighted graphs.

▶ **Theorem 10.** *For any $\varepsilon > 0, \rho \in (-1, 0), \gamma > 0$, there is a sufficiently large alphabet size $L \in \mathbb{N}$ and a reduction from regular Affine Unique Games instances $\Lambda = (\mathcal{U}, \mathcal{V}, \mathcal{E}, \Pi, [L])$ to weighted multigraphs $G = (V, E)$ with the following properties:*

[4] Without loss of generality, we assume that weights of edges sum up to 1 here.

- Completeness: *If* $\mathrm{Opt}(\Lambda) \geq 1-\gamma$*, then* $\mathrm{MSVC}(G) \leq \left(\frac{1}{3-\rho} + \varepsilon + 3\gamma\right)|V|$.
- Soundness: *If* $\mathrm{Opt}(\Lambda) \leq \gamma$*, then for every* $r \in [0,1]$*,* G *is* $(r, \Gamma_\rho(r) - \varepsilon)$*-dense.*

Moreover, the running time of the reduction is polynomial in $|\mathcal{U}|, |\mathcal{V}|, |\mathcal{E}|$*, and exponential in* L*. If we use* D *to denote the degree of the regular Unique Games instance, then the weights of edges in* G *belong to the set* $\left\{\left(\frac{1+\rho}{4}\right)^i \left(\frac{1-\rho}{4}\right)^{L-i}\right\}_{i=0}^{L}$*. The size of* $|V|$ *is at least* 2^L*. Finally, the output graph* G *is also regular, in the following two senses. First, if we consider* G *as an unweighted graph, every vertex is of degree* D^2*. The graph* G *is also regular in the weighted sense, i.e., the value* $W_E(u, N(u))$ *is uniform across all* $u \in V$*, and it equals* $D^2 2^{-L+1}$*.*

Proof. Let $\nu\colon \{0,1\}^2 \to [0,1]$ be the probability distribution over correlated uniformly distributed bits with negative correlation coefficient $\rho < 0$. In other words, we have

$$\nu(0,0) = \nu(1,1) = \frac{1+\rho}{4}, \quad \nu(0,1) = \nu(1,0) = \frac{1-\rho}{4}.$$

Let us now describe how the multigraph G can be constructed from Λ. We define the vertex set of G to be $V = \mathcal{V} \times \{0,1\}^L = \{(v,x) \mid v \in \mathcal{V}, x \in \{0,1\}^L\}$. In particular, for every vertex $v \in \mathcal{V}$ we create 2^L vertices of G, which we identify with L-bit strings in $\{0,1\}^L$. We also write v^x for a vertex (v,x) of the graph G. The edges of G are constructed in the following way. For every $u \in \mathcal{U}$, and for every two $v_1, v_2 \in N(u)$, we create an edge between vertices v_1^x, v_2^y with weight

$$\nu^{\otimes L}(x \circ \pi_{e_1}, y \circ \pi_{e_2}), \quad \text{where } e_1 = (u, v_1), \quad e_2 = (u, v_2).$$

Expressed formally, the edge set E is

$$E = \{(e_1^x, e_2^y) \mid e_1 = (u, v_1), e_2 = (u, v_2), u \in \mathcal{U}, v_1, v_2 \in \mathcal{V}, x, y \in \{0,1\}^L\}.$$

The number of vertices in G is $|\mathcal{V}|2^L$, and the number of edges is $|\mathcal{V}|D^2 2^L$, so the construction is indeed polynomial in $|\mathcal{U}|, |\mathcal{V}|$ and $|\mathcal{E}|$, and exponential in L. Also, since $\mathcal{V} \neq \emptyset$ we have $|V| \geq 2^L$, and the weights of the edges indeed belong to the set specified in the statement of the theorem. Finally, the total weight of edges incident upon each vertex v^x is the same for any v^x, and since $W_E(E) = D^2|\mathcal{V}|$, we have that $w_E(v^x, N(v^x)) = 2D^2|\mathcal{V}|\frac{1}{|V|} = D^2 2^{-L+1}$ for all $v^x \in V$.

We are using the same reduction[5] as the one used in Theorem 3.1. from [5], and the only difference is that we are starting from Affine Unique Games instead of (general) Unique Games. Since we are using the same reduction and Affine Unique Games are subsumed by the Unique Games, our graph G satisfies the same soundness property as the one expressed by Theorem 3.1. in [5], and this is exactly the soundness property stated above. Hence, we only need to show completeness.

For the completeness case let us assume $\mathrm{Opt}(\Lambda) \geq 1-\gamma$. Therefore, there is a labelling $z\colon \mathcal{U} \sqcup \mathcal{V} \to \mathbb{Z}_L$ such that $\mathrm{Val}_z(\Lambda) \geq 1-\gamma$. In particular, there is $\hat{\mathcal{E}} \subseteq \mathcal{E}, |\hat{\mathcal{E}}| \geq (1-\gamma)|\mathcal{E}|$, such that for each $e = (u,v) \in \hat{\mathcal{E}}$ we have $z(u) - z(v) = c_e$. Let us use $\hat{E} \subseteq E$ to denote the set

$$\hat{E} := \{(e_1^x, e_2^y) \in E \mid e_1, e_2 \in \hat{\mathcal{E}}\}.$$

Observe that $|\hat{E}| \geq |E| \cdot (1-2\gamma)$. Since the complement of $\hat{E}$ is of small fractional size, i.e., smaller than 2γ, in the analysis we will focus on cover times of edges in $\hat{E}$, and we will

[5] This is the same as the Max-Cut hardness reduction in [11]. Same reduction and soundness result also appeared in [4], albeit with biased bits.

trivially upper bound the cover time of edges in $\hat{E}^c$ by $|V|$. In particular, let us denote with $\hat{G}$ the graph $\hat{G} = (V, \hat{E})$ and find an ordering σ such that $SVC_{\hat{G}}(\sigma) \leq (\frac{1}{3-\rho} + \varepsilon + \gamma)|V|$. As discussed this would then give us the stated completeness

$$SVC_G(\sigma) \leq \left(\frac{1}{3-\rho} + \varepsilon + 3\gamma\right)|V|,$$

by bounding the cover time of edges in $\hat{E}^c$ by $|V|$.

Before explaining how σ is constructed, let us first introduce some notation. We use $z_1, \ldots, z_L \colon \mathcal{U} \to \mathbb{Z}_L$ to denote the mappings defined by

$$z_i(u) \colon = z(u) + i, \quad \text{for } i \in [L].$$

Let us then define sets $F_i^0, F_i^1 \subseteq V$, as the sets in which, for every $v \in \mathcal{V}$, inside the long code (v, x) we fix the $z_i(v)$-th coordinate to 0 or 1, respectively. In particular, we have

$$F_i^0 = \left\{(v, x) \in V \mid x_{z_i(v)} = 0\right\}, \quad F_i^1 = \left\{(v, x) \in V \mid x_{z_i(v)} = 1\right\}.$$

Intuitively, the sets F_i^0 (or F_i^1) for a fixed i fix the values at the coordinates in which labels "agree". Then, we use the sets F_i^0 and F_i^1 to construct ordering inductively. First, we define the ordering on $C_{L-1} = F_1^0 \cap F_2^0 \cap \ldots F_{L-1}^0$, then using this ordering we define ordering on $C_{L-2} = F_1^0 \cap F_2^0 \cap \ldots F_{L-2}^0$, and so on until we construct an ordering on $C_1 = F_1^0$ and finally on C_0 which we define to be $C_0 := V$. As we are defining orderings on $C_i, i = 0, \ldots, L-1$, we will be expressing an upper bound T_i for the average time edges E_i with both endpoints endpoints in C_i are covered by the ordering. Before discussing our ordering, let us make an observation that $|C_i| = 2 \cdot |C_{i+1}|$, since C_i has one more free coordinate for each $v \in \mathcal{V}$.

We discuss the ordering for C_{L-1} first. Before that, let us remark that the particular ordering and the cost of covering edges in C_{L-1} will be inconsequential for the final value that we get in this theorem. The main reason we discuss this case here is because we believe it will be a good preparation for discussing the inductive step that will follow. Let us first normalize the weights of edges in E_{L-1} so that they sum up to 1. In the first step, we iterate through $v \in \mathcal{V}$ in a random order[6], and pick $(v, x) \in C_{L-1}$ such that[7] $x_{z_L(v)} = 0$. Then, we iterate through $v \in \mathcal{V}$ in a random order and pick the remaining vertex at each (v, x), i.e., the vertex with $x_{z_L(v)} = 1$. Let us upper bound the average time an edge $e \in E_{L-1}$ with both endpoints in C_{L-1} is visited by this schedule. Observe that we spent $\frac{1}{2}|C_{L-1}|$ time in the first step, and $\frac{1}{2}|C_{L-1}|$ in the second step. Thus, if an edge with both endpoints in C_{L-1} has at least one endpoint with a label 0 at $x_{z_L(v)}$, then this point will be picked in the first step on average by the time $\frac{1}{4}|C_{L-1}|$. Otherwise, if the edge e has both endpoints v_1^x, v_2^y picked in the second step, i.e. $x_{Z_L(v_1)} = 1, y_{Z_L(v_2)} = 1$, then it will be picked on average by the time $\frac{3}{4}|C_{L-1}|$. Since the weight of edges from E_{L-1} picked in the first step is $\nu(0,0) + \nu(0,1) + \nu(1,0) = \frac{3-\rho}{4}$, and the weight of the remaining edges that we consider is $\nu(1,1) = \frac{1+\rho}{4}$, the average cover time is

$$T_{L-1} = \frac{3-\rho}{4} \cdot \frac{1}{4}|C_{L-1}| + \frac{1+\rho}{4}\frac{3}{4}|C_{L-1}| = \frac{3+\rho}{8}|C_{L-1}|.$$

We observe that this also shows that there is an ordering σ_{L-1} which covers an edge in E_{L-1} on average by the time T_{L-1}.

[6] As we have said, the value obtained in the first step is not relevant as it will be seen later. Hence, we can also choose to visit $v \in \mathcal{V}$ in any fixed order in which case we can also use a trivial upper bound of $|V_{L-1}|$ on T_{L-1}.

[7] Due to symmetry it is not important whether we pick $x_{z_L(v)} = 0$ or $x_{z_L(v)} = 1$ in the first iteration, as long as we keep that choice fixed.

Let us now fix $i = 0, \ldots, L-2$, and assume that we have an ordering σ_{i+1} of vertices in C_{i+1} such that the edges in E_{i+1} are covered by the time T_{i+1} on average, and let us use this procedure to construct an ordering of the vertices in C_i and derive a suitable upper bound on T_i. We assume that the weights of the edges E_i are normalized so that they sum up to 1. The ordering in C_i works as follows. First, using σ_{i+1} we visit vertices in $C_{i+1} = C_i \cap F_i^0$. The total weight of the edges with both endpoints in $C_i \cap F_i^0$ is $\nu(0,0)$, and they are covered by σ_{i+1} until T_i on average. Hence, the cost for these edges is

$$T_{i+1} \cdot \nu(0,0). \tag{4}$$

Furthermore, during this pass, we have also visited all the edges with one endpoint in $C_i \cap F_i^0$ and another endpoint in $C_i \cap F_i^1$, and their total weight is $\nu(0,1) + \nu(1,0)$. Also, these covers are disjoint (each one of these edges will be visited only once in the first pass). Since the starting Unique Games instance was regular and we removed at most 2γ edges, the edges will be covered by the time

$$\frac{1+2\gamma}{2}|C_{i+1}| \tag{5}$$

at most. Hence, the cost for these edges is

$$(\nu(0,1) + \nu(1,0))\frac{1+2\gamma}{2}|C_{i+1}|. \tag{6}$$

Finally, we pass through the vertices in $C_i \cap F_i^1$. The graph induced by this vertex set is actually isomorphic to $C_{i+1} = C_i \cap F_i^0$, and hence we can once again use the ordering σ_{i+1}. Then, the edges in this graph are visited on average by the time

$$|C_{i+1}| + T_{i+1},$$

where the $|C_{i+1}|$ term is due to the delay coming from the first pass. Hence, the cost of these edges is at most

$$\nu(1,1)(|C_{i+1}| + T_{i+1}). \tag{7}$$

Adding up (4),(6) and (7) we get that

$$T_i \le \frac{1+\rho}{4}T_{i+1} + \frac{1-\rho}{2}\frac{1+2\gamma}{2}|C_{i+1}| + \frac{1+\rho}{4} \cdot (|C_{i+1}| + T_{i+1}). \tag{8}$$

If we let $t_i = T_i/|C_i|$ and divide both sides by $|C_i| = 2|C_{i+1}|$, we can write (8) as

$$t_i \le \frac{1+\rho}{8}t_{i+1} + \frac{1-\rho}{4}\frac{1+2\gamma}{2} + \frac{1+\rho}{4}\left(\frac{1+t_{i+1}}{2}\right),$$

which can be simplified to

$$t_i \le \frac{1}{4} + \frac{1+\rho}{4}t_{i+1} + \frac{1-\rho}{4}\gamma. \tag{9}$$

Let us show that $t_i \le \frac{1}{3-\rho} + \gamma + 2^{-L+i}$ as follows. Let us define $r_i = t_i - \gamma - \frac{1}{3-\rho}$. By substituting $t_i = \frac{1}{3-\rho} + \gamma + r_i$ into (9) we obtain

$$\frac{\gamma}{2} + r_i \le \frac{1+\rho}{4}r_{i+1}. \tag{10}$$

Since $\rho \in (-1, 0)$ and $\gamma > 0$ we have

$$r_i \leq \frac{1}{2} r_{i+1}. \tag{11}$$

Hence, since by calculation for T_{L-1} we have $r_{L-1} \leq \frac{1}{2}$, which with (11) implies that $r_i \leq 2^{-L+i}$, and therefore $t_0 \leq \frac{1}{3-\rho} + \gamma + 2^{-L}$. By letting L be large enough so that $2^{-L} \leq \varepsilon$ and recalling that $t_0 = T_0/|V|$ we get

$$T_0 \leq (\frac{1}{3-\rho} + \gamma + \varepsilon)|V|,$$

which is what we wanted to prove. ◀

This reduction outputs a weighted graph. In the full version of this paper we will show how this weighted graph can be transformed into an unweighted graph with *essentially* the same properties using a polynomial time reduction. For now, let us briefly discuss how the soundness and completeness properties stated in the theorem above are useful for studying Min Sum Vertex Cover.

For completeness, we will get that $\mathrm{MSVC}(G) \leq \left(\frac{1}{3-\rho} + \varepsilon + 3\gamma\right)|V|$. On the other hand, in the soundness case we have that for any ordering σ and given any $\eta > 0$ we have

$$\begin{aligned}\mathrm{SVC}_G(\sigma) &= \sum_{t=1}^{|V|} w(\sigma([t])^c, \sigma([t])^c) \geq \sum_{t=1}^{n-\lceil \eta n \rceil} w(\sigma([t])^c, \sigma([t])^c) \geq \sum_{t=1}^{n-\lceil \eta n \rceil} \Gamma_\rho(1 - t/n) - \varepsilon \\ &= \left(\int_0^{1-\eta} \Gamma_\rho(1-r)dr - \varepsilon\right) \cdot |V| + O(1) = \left(\int_\eta^1 \Gamma_\rho(r)dr - \varepsilon\right) \cdot |V| + O(1).\end{aligned}$$

Hence, by letting $\eta \to 0, \gamma \to 0, \varepsilon \to 0, |V| \to \infty$, we get an inapproximability ratio of

$$\frac{\int_0^1 \Gamma_\rho(r)dr}{\frac{1}{3-\rho}}.$$

This expression is minimized for $\rho \approx -0.52$, for which the inapproximability ratio is approximately 1.014.

Numerical simulations show that the best ratio we can get with these techniques is 1.014, and it is obtained for $\rho = -0.52$.

4 Approximating Min Sum Vertex Cover on Regular Graphs

In this section we will revisit an approximation algorithm for Minimum Sum Vertex Cover on regular graphs introduced in [9], in Theorem 11. The authors in that work did not explicitly state the approximation ratio obtained by that algorithm, since their primary interest was showing that 4/3-approximation achieved by the greedy algorithm can be beaten by more advanced techniques.

We will here give an explicit constant, also taking into account progress in the approximation of the so called Max-k-VC problem, which is used in that approach, and for which better algorithms exist since the publication of the aforementioned article.

Before discussing the algorithm, let us define the Max-k-VC problem. In this problem a graph $G = (V, E)$ is given as an input, and the goal is to find $S \subseteq V, |S| = k$, such that $w(S, V)$ is as big as possible. Austrin, Benabbas and Georgiou [3] show that Max-2-Sat with a bisection constraint, that is, Max-2-Sat in which admissible assignments have exactly half of the variables set to 1, and the other half to 0, can be approximated within $\alpha_{LLZ} \approx 0.9401$. Let us remark that α_{LLZ} is the optimal[8] approximation ratio for Max-2-Sat problem [14, 2]. Since this problem subsumes Max-k-VC when $k = n/2$, we can approximate Max-$n/2$-VC within $\alpha_{LLZ} \approx 0.9401$.

Let us also recall the following two facts for regular graphs:

- The greedy algorithm on regular graphs covers edges on average by the time $\frac{1}{3}|V|$,
- The optimal solution covers an edge on average by the time at least $\frac{1}{4}|V|$.

Let us now discuss the algorithm introduced in [9]. We will closely follow the argument outlined there. Let $\varepsilon > 0$ be some constant that we will fix later. In case the optimal solution covers an edge by the time $(\frac{1}{4} + \varepsilon)|V|$ later, the greedy algorithm approximates the optimal value within a factor of

$$\frac{1/3}{\frac{1}{4} + \varepsilon} = \frac{4}{3 + \frac{3}{4}\varepsilon}.$$

Otherwise, the optimal solution covers an edge on average at the time $(\frac{1}{4} + \delta)|V|$, for some $\delta \in (0, \varepsilon)$. In this case, we have the following lemma.

▶ **Lemma 11.** *Let $G = (V, E)$ be a regular graph, let $n := |V|$, and let the optimal solution of Minimum Sum Vertex Cover be $(\frac{1}{4} + \delta)|V|$. Then the optimal solution covers at least $(1 - \sqrt{\delta})$ fraction of edges in the first $n/2$ steps.*

Proof. Let us denote the degree of the graph with $D \in \mathbb{N}$, and with m the number of edges $m = nD/2$. We argue by contradiction, and assume that the optimal solution covers less than $(1 - \sqrt{\delta})$ fraction of edges in the first $n/2$ steps. Let us use $u_i, i = 1, \ldots, n$, to denote the number of uncovered edges at the time step i, and let $s := u_{n/2}$. Then by assumption $s > \sqrt{\delta}m$. Furthermore, the value of the minimum sum vertex cover is $\frac{1}{m}\sum_{i=1}^{n} u_i$. Let us show that $\frac{1}{m}\sum_{i=1}^{n} u_i > (\frac{1}{4} + \delta)n$ yielding a contradiction to the assumption that the optimal solution of Minimum Sum Vertex Cover is $(\frac{1}{4} + \delta)n$.

Let us use $c_i = u_i - u_{i-1}$ to denote the number of additionally covered edges at step i. Since we are considering the optimal solution to MSVC, the sequence c_i is non-increasing (otherwise changing the order would yield a smaller solution). Furthermore, let us use c to denote $c_{n/2}$.

Now, assuming that $c_{n/2} = c$, $u_{n/2} = s$, let us calculate the smallest possible value of MSVC. We know that after i steps, we can cover at most $i \cdot D$ edges (this happens if all the edges chosen are disjoint). Furthermore, since we assumed that after $n/2$ steps we leave s edges uncovered, and since $c = c_{n/2}$ and c is non-increasing, we have that at the step i we leave at least $s + (n/2 - i) \cdot c$ edges uncovered. This shows that

$$u_i \geq \max\left(\frac{nD}{2} - iD, s + (n/2 - i) \cdot c, 0\right), i \in [n].$$

In particular, the right hand side is a maximum of three linear functions, and therefore, the following scenario for covering the edges will lower bound u_i. In the first t fraction of steps, edges get covered at the optimal rate (at each step we cover D new edges), where t is a parameter calculated later. Then, after t fraction of steps, we cover c edges, until we cover all the edges[9].

[8] Assuming the Unique Games Conjecture

[9] In the last step we might cover less than c edges, but we will ignore this case for the sake of simplicity.

Since we spend t fraction of time covering D edges in each step, the cost of edges covered in this time is

$$\frac{1}{m}\sum_{i=1}^{t\cdot n}\frac{Dn}{2} - iD \geq nt(1-t).$$

The remaining time $x \cdot n$, for some $x \in (0,1)$, is spent on covering c edges at each step. Since after x fraction of steps we covered all the edges, we have

$$m = t \cdot n \cdot D + x \cdot n \cdot c,$$

and since $m = nD/2$ we have that

$$x = \frac{D}{2} \cdot \frac{1-2t}{c}. \tag{12}$$

Hence, the average cost of edges incurred in the remaining time is

$$\frac{1}{m} x \cdot n \cdot (m - t \cdot n \cdot D)\frac{1}{2} = xn\left(\frac{1}{2} - t\right),$$

which with (12) yields the cost

$$\frac{D}{2} \cdot \frac{1-2t}{c} n \left(\frac{1}{2} - t\right) = n\frac{D}{4c}(1-2t)^2.$$

Hence, the total cost is

$$nt(1-t) + n\frac{D}{4c}(1-2t)^2 \tag{13}$$

Let us now calculate the value of t in terms of s and c. We use the fact that after t fraction of steps we covered $t \cdot n \cdot D$ edges, and after $n/2$ steps we covered $m - s$ edges. Since we are covering c edges at each step $i \in [tn, n/2]$ we have that

$$m - s = tnD + \left(\frac{n}{2} - tn\right) \cdot c,$$

and from here we get

$$t = \frac{m - s - \frac{n \cdot c}{2}}{n \cdot D - n \cdot c} = \frac{1}{2} - \frac{s}{n(D-c)}.$$

Replacing this in (13) we get that the total cost is at least

$$n\left(\frac{1}{4} - \frac{s^2}{n^2(D-c)^2} + \frac{D}{4c}\left(2\frac{s}{n(D-c)}\right)^2\right)$$

which reduces to

$$n\left(\frac{1}{4} + \frac{s^2}{n^2}\frac{1}{(D-c)c}\right)$$

Now, by our contradiction hypothesis we have $s > \sqrt{\delta} m$, and $\frac{1}{(D-c)c} \geq 4\frac{1}{D^2}$ (since $c \in (1, D)$), we have that the total cover time is strictly greater than

$$n\left(\frac{1}{4} + \frac{\delta m^2}{n^2}\frac{4}{D^2}\right) = n\left(\frac{1}{4} + \delta\right),$$

which contradicts the fact that the optimal solution to MSVC on the graph G has value $n(\frac{1}{4} + \delta)$. This concludes our proof. ◀

In [9] it is claimed that in the setup of Lemma 11, the optimal solution covers $(1-\delta)$ fraction of edges. However, this is not correct, and we illustrate that with a counterexample provided in the appendix. Nevertheless, this does not greatly change the conclusion in [9], as using the correct version of the lemma just replaces one unspecified constant below 4/3 by another unspecified constant below 4/3.

Let us now fix $k = n/2$, and use the Max-k-VC algorithm. This will give us a set $S \subseteq V$ such that $w(S, V) \geq \alpha_{LLZ}(1-\sqrt{\delta})$. We next consider the following ordering of vertices in V and calculate Minimum Sum Vertex Cover for it. We first pick vertices from S in a random order, and then take the remaining vertices in random order as well. Then, the vertices in $N(S, S)$ are covered by the time $|V|/4$ in expectation, while the remaining vertices are covered by the time $\frac{|V|}{2} + \frac{1}{3}\frac{|V|}{2}$. Hence, we can find an ordering for which Sum Vertex Cover has value

$$w(S,V)\frac{|V|}{4} + w(S^c, S^c)(\frac{|V|}{2} + \frac{|V|}{6}) \leq \alpha_{LLZ}(1-\sqrt{\delta})\frac{|V|}{4} + \Big(1 - \alpha_{LLZ}(1-\sqrt{\delta})\Big) \cdot \frac{2|V|}{3}.$$

We can simplify this expression as

$$\left(\frac{-5\alpha_{LLZ} + 5\alpha_{LLZ}\sqrt{\delta}}{12} + \frac{2}{3}\right)|V|.$$

Hence, we get an approximation ratio of

$$\frac{\frac{-5\alpha_{LLZ}+5\alpha_{LLZ}\sqrt{\delta}}{12} + \frac{2}{3}}{\frac{1}{4} + \delta}.$$

In conclusion, for fixed ε we have that the approximation ratio is given as

$$\max\left(\frac{4}{3+\frac{3}{4}\varepsilon}, \sup_{\delta\in(0,\varepsilon]} \frac{\frac{-5\alpha_{LLZ}+5\alpha_{LLZ}\sqrt{\delta}}{12} + \frac{2}{3}}{\frac{1}{4} + \delta}\right).$$

Optimizing over different values of ε gives us that the approximation ratio of this algorithm is approximately 1.225.

References

1. Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, May 1998. doi:10.1145/278298.278306.
2. Per Austrin. Balanced max 2-sat might not be the hardest. In *Proceedings of the 39th Annual ACM Symposium on Theory of Computing, San Diego, California, USA, June 11-13, 2007*, pages 189–197, 2007. doi:10.1145/1250790.1250818.
3. Per Austrin, Siavosh Benabbas, and Konstantinos Georgiou. Better balance by being biased: A 0.8776-approximation for max bisection. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 277–294, 2013. doi:10.1137/1.9781611973105.21.
4. Per Austrin, Subhash Khot, and Muli Safra. Inapproximability of vertex cover and independent set in bounded degree graphs. *Theory Comput.*, 7(1):27–43, 2011. doi:10.4086/toc.2011.v007a003.
5. Per Austrin and Aleksa Stankovic. Global cardinality constraints make approximating some max-2-csps harder. In Dimitris Achlioptas and László A. Végh, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2019, September 20-22, 2019, Massachusetts Institute of Technology, Cambridge, MA, USA.*, volume 145 of *LIPIcs*, pages 24:1–24:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019. doi:10.4230/LIPIcs.APPROX-RANDOM.2019.24.

6 Nikhil Bansal and Subhash Khot. Optimal long code test with one free bit. In *50th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2009, October 25-27, 2009, Atlanta, Georgia, USA*, pages 453–462. IEEE Computer Society, 2009. doi:10.1109/FOCS.2009.23.

7 Uri Barenholz, Uriel Feige, and David Peleg. Improved approximation for min-sum vertex cover. *Technical report, MCS06-07, Computer Science and Applied Mathematics*, 2006.

8 Vaggos Chatziafratis, Neha Gupta, and Euiwoong Lee. Inapproximability for local correlation clustering and dissimilarity hierarchical clustering. *CoRR*, abs/2010.01459, 2020. arXiv:2010.01459.

9 Uriel Feige, László Lovász, and Prasad Tetali. Approximating min sum set cover. *Algorithmica*, 40(4):219–234, 2004. doi:10.1007/s00453-004-1110-5.

10 Subhash Khot. On the power of unique 2-prover 1-round games. In *Proceedings on 34th Annual ACM Symposium on Theory of Computing, May 19-21, 2002, Montréal, Québec, Canada*, pages 767–775, 2002. doi:10.1145/509907.510017.

11 Subhash Khot, Guy Kindler, Elchanan Mossel, and Ryan O'Donnell. Optimal inapproximability results for MAX-CUT and other 2-variable csps? *SIAM J. Comput.*, 37(1):319–357, 2007. doi:10.1137/S0097539705447372.

12 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2-\varepsilon$. In *18th Annual IEEE Conference on Computational Complexity (Complexity 2003), 7-10 July 2003, Aarhus, Denmark*, page 379, 2003. doi:10.1109/CCC.2003.1214437.

13 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. doi:10.1016/j.jcss.2007.06.019.

14 Michael Lewin, Dror Livnat, and Uri Zwick. Improved rounding techniques for the MAX 2-sat and MAX DI-CUT problems. In *Integer Programming and Combinatorial Optimization, 9th International IPCO Conference, Cambridge, MA, USA, May 27-29, 2002, Proceedings*, pages 67–82, 2002. doi:10.1007/3-540-47867-1_6.

15 Prasad Raghavendra. Optimal algorithms and inapproximability results for every csp? In *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*, pages 245–254, 2008. doi:10.1145/1374376.1374414.

16 Aleksa Stankovic. On regularity of max-csps and min-csps. *Inf. Process. Lett.*, 176:106244, 2022. doi:10.1016/j.ipl.2022.106244.

A Addressing Argument in Theorem 11 in [9]

Let $\delta > 0$. We construct a 2-regular graph $G = (V, E)$ with $|V| = n, |E| = m$, such that the minimum sum vertex cover has value

$$\left(\frac{1}{4} + \delta\right) n,$$

while any set $S \subseteq V, |S| = \frac{n}{2}$, satisfies $|w(S, V)| \leq (1 - \sqrt{\delta})$. This shows that the factor $(1 - \sqrt{\delta})$ in Lemma 11 can not be replaced by a sharper $(1 - \delta)$, as it was done in [9].

Let $t = (\frac{1}{2} - 3\sqrt{\delta})n$ and $s = 2\sqrt{\delta} \cdot n$, where $n \in \mathbb{N}$ is chosen such that $t, s \in \mathbb{N}$ (we also approximate $\sqrt{\delta}$ by a rational number), and such that t is even. Then, we construct the graph G by taking $t/2$ disjoint copies of $K_{2,2}$ and s disjoint copies of K_3. Let V_1 be the set composed of only "the left sides" of $t/2$ disjoint copies of $K_{2,2}$, V_2 the set composed of only "the right sides" of $t/2$ disjoint copies of $K_{2,2}$, and let V_3 be composed of the vertices from s disjoint copies of K_3.

Then, the optimal solution for MSVC will work in the following three stages:

- *Stage 1:* Pick vertices from V_1 in any order.
- *Stage 2:* Pick one vertex from each K_3 in V_3.
- *Stage 3:* Pick another vertex from each K_3 in the set V_3.

It is clear that this is the minimum sum vertex cover. The total cost is then split into the following costs:

- Edges covered in the first stage. In this case we pick $t \cdot 2$ edges, and an edge is picked on average at the time $t/2$, so the cost is

$$\frac{t}{2} \cdot 2 \cdot t = t^2.$$

- Edges covered in the second stage. We pick $s \cdot 2$ edges, and each edge is picked on average at the time $t + s/2$, where the factor t exists because this step happens after the first stage. We have the cost of

$$2 \cdot s(t + s/2) = 2 \cdot s \cdot t + s^2.$$

- Edges covered in the third stage. We pick s edges, and each edge is picked on average at the time $t + s + s/2$, where the factor $t + s$ exists because this step happens after the second stage. We have the cost of

$$s(t + s + s/2) = st + s^2 + s^2/2.$$

Hence, the total cost is

$$t^2 + 2 \cdot s \cdot t + s^2 + st + s^2 + s^2/2 = t^2 + 3st + \frac{5s^2}{2}.$$

Now, recalling that $t = (\frac{1}{2} - 3\sqrt{\delta})n$ and $s = 2\sqrt{\delta} \cdot n$, we have that the cost is

$$\begin{aligned} n^2 \cdot \left(\frac{1}{4} - 3\sqrt{\delta} + 9\delta\right) + 3 \cdot 2\sqrt{\delta} \cdot n \cdot \left(\frac{1}{2} - 3\sqrt{\delta}\right) \cdot n + \frac{5 \cdot 4\delta \cdot n^2}{2} \\ = \frac{1}{4}n^2 - 3\sqrt{\delta}n^2 + 9\delta n^2 + 3\sqrt{\delta}n^2 - 18\delta n^2 + 10\delta n^2 \\ = \frac{1}{4}n^2 + \delta n^2. \end{aligned}$$

Now, since our graph is 2-regular graph on $n = 2t + 3s$ vertices, we have that $m = n$, and we can write the total cost as

$$n\left(\frac{1}{4} + \delta\right),$$

as claimed. It remains to show that for any set $S \subseteq V$ with $|S| = n/2$ we have

$$|E(S, V)| \leq (1 - \sqrt{\delta}) \cdot m.$$

It is obvious that the worst case S is exactly the set of vertices picked in the first $n/2$ steps in the algorithm above. In this case, the number of edges not covered is $s/2$, since after $n/2$ steps we are left with one edge uncovered in exactly half of the K_3 triangles. Hence, the number of uncovered edges is

$$\frac{s}{2} = \sqrt{\delta} \cdot n = \sqrt{\delta} \cdot m,$$

and hence

$$|E(S, V)| \leq (1 - \sqrt{\delta}) \cdot m,$$

as required.