



UNIVERSITY OF LEEDS

This is a repository copy of *Parameterized Algorithms for Minimum Sum Vertex Cover*.

White Rose Research Online URL for this paper:

<https://eprints.whiterose.ac.uk/208305/>

Version: Accepted Version

Proceedings Paper:

Aute, S. and Panolan, F. orcid.org/0000-0001-6213-8687 (2024) Parameterized Algorithms for Minimum Sum Vertex Cover. In: Soto, J.A. and Wiese, A., (eds.) LATIN 2024: Theoretical Informatics 16th Latin American Symposium, Puerto Varas, Chile, March 18–22, 2024, Proceedings, Part II. LATIN 2024: Latin American Symposium on Theoretical Informatics, 18-22 Mar 2024, Puerto Varas, Chile. Lecture Notes in Computer Science, 14579 . Springer , pp. 193-207. ISBN 978-3-031-55601-2

https://doi.org/10.1007/978-3-031-55601-2_13

© The Author(s). This version of the article has been accepted for publication, after peer review (when applicable) and is subject to Springer Nature's AM terms of use (<https://www.springernature.com/gp/open-research/policies/accepted-manuscript-terms>), but is not the Version of Record and does not reflect post-acceptance improvements, or any corrections. The Version of Record is available online at: http://dx.doi.org/10.1007/978-3-031-55601-2_13

Reuse

Items deposited in White Rose Research Online are protected by copyright, with all rights reserved unless indicated otherwise. They may be downloaded and/or printed for private study, or other acts as permitted by national copyright laws. The publisher or other rights holders may allow further reproduction and re-use of the full text version. This is indicated by the licence information on the White Rose Research Online record for the item.

Takedown

If you consider content in White Rose Research Online to be in breach of UK law, please notify us by emailing eprints@whiterose.ac.uk including the URL of the record and the reason for the withdrawal request.



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

Parameterized Algorithms for Minimum Sum Vertex Cover

Shubhada Aute¹[0009-0000-2964-0368] and Fahad Panolan²[0000-0001-6213-8687]

¹ Indian Institute of Technology Hyderabad, Sangareddy, India
aute.shubhada@gmail.com

² School of Computing, University of Leeds, Leeds, UK
fahad.panolan@gmail.com

Abstract. Minimum sum vertex cover of an n -vertex graph G is a bijection $\phi : V(G) \rightarrow [n]$ that minimizes the cost $\sum_{\{u,v\} \in E(G)} \min\{\phi(u), \phi(v)\}$. Finding a minimum sum vertex cover of a graph (the MSVC problem) is NP-hard. MSVC is studied well in the realm of approximation algorithms. The best-known approximation factor in polynomial time for the problem is $16/9$ [Bansal, Batra, Farhadi, and Tetali, SODA 2021]. Recently, Stankovic [APPROX/RANDOM 2022] proved that achieving an approximation ratio better than 1.014 for MSVC is NP-hard, assuming the Unique Games Conjecture. We study the MSVC problem from the perspective of parameterized algorithms. The parameters we consider are the size of a minimum vertex cover and the size of a minimum clique modulator of the input graph. We obtain the following results.

- MSVC can be solved in $2^{2^{O(k)}} n^{O(1)}$ time, where k is the size of a minimum vertex cover.
- MSVC can be solved in $f(k) \cdot n^{O(1)}$ time for some computable function f , where k is the size of a minimum clique modulator.

Keywords: FPT · Vertex Cover · Integer Quadratic Programming

1 Introduction

A vertex cover in a graph is vertex subset such that each edge has at least one endpoint in it. Finding a vertex cover of minimum size is NP-complete, and it is among the renowned 21 NP-complete problems proved by Karp in 1972 [24]. Since then, it has been extensively studied in both the fields of approximation algorithms and parameterized algorithms. The approximation ratio 2 of minimum vertex cover is easily achievable using any maximal matching of the graph and it is optimal assuming Unique Games Conjecture [25]. The best-known FPT algorithm for vertex cover has running time $O(1.2738^k + kn)$ [8], where k is the minimum vertex cover size. In this paper, we study the well-known MINIMUM SUM VERTEX COVER, defined below. For $n \in \mathbb{N}$, $[n] = \{1, 2, \dots, n\}$.

Definition 1. Let G be an n -vertex graph, and let $\phi : V(G) \rightarrow [n]$ be a bijection. The weight (or cover time or cost) of an edge $e = \{u, v\}$ is $w(e, \phi) =$

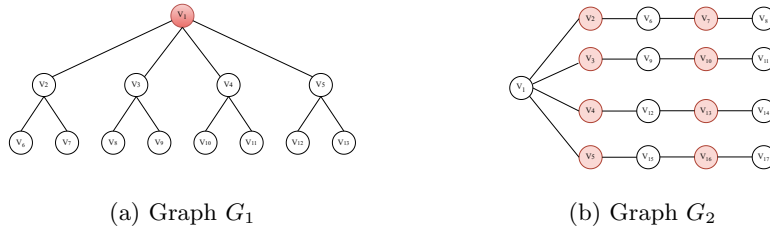


Fig. 1: Counter example

$\min\{\phi(u), \phi(v)\}$. The cost of the ordering ϕ for the graph G is defined as,

$$\mu_G(\phi) = \sum_{e \in E(G)} w(e, \phi).$$

The objective of MINIMUM SUM VERTEX COVER (MSVC) problem is to find an ordering ϕ that minimizes $\mu_G(\phi)$ over all possible orderings. MINIMUM SUM VERTEX COVER came up in the context of designing efficient algorithms for solving semidefinite programs [7]. It was first studied by Feige et al. [12], though their main focus was on MINIMUM SUM SET COVER (MSSC), which is a generalization of the MSVC on hypergraphs.

As MSVC requires the sum of the weights of the edges to be minimized, it is natural to think of prioritizing vertices based on their degree. But this isn't always an optimal approach. For instance, v_1 is the vertex of the maximum degree in the graph G_1 as shown in Figure 1a. Consider an ordering ψ with v_1 in the first position. For position 2 onwards, any vertex from $G_1 - v_1$ can cover at most two edges. So, the cost of such an ordering is at least 32. However, if the vertices v_2, v_3, v_4, v_5 take the first four locations in an ordering, say ϕ , then cost of ϕ is 30. Though greedy isn't an optimal strategy, it is interesting to find the graphs for which preferring a vertex of highest degree at each location from the remaining graph yields an optimal solution for MSVC. Mohan et al. gave a sufficiency condition for graphs admitting greedy solution as optimal [28]. Interestingly, the greedy approach performs no worse than 4 times the optimum cost. However, certain bipartite graphs yield a solution from the greedy algorithm that is precisely four times the optimal solution for MSVC [12]. The greedy algorithm achieves factor 4 approximation for MSSC on hypergraphs as well [3].

Both the problems MSVC and MSSC are well-studied in the realm of approximation algorithms. An improvement over the greedy algorithm is a 2-factor approximation for MSVC using linear programming [12]. The approximation ratio for MSVC was further improved to 1.999946 [4]. The best approximation ratio for MSVC is 16/9, which was achieved using a generalized version of MSSC, called the generalized min-sum set cover (GMSSC) [2]. Input to GMSSC is a hypergraph $H = (V, E)$ in which every hyperedge has a covering requirement k_e where $k_e \leq |e|$. The first location where k_e number of vertices of an edge e has appeared in the ordering is the cover time of e . The GMSSC problem is the

problem of finding an ordering of vertices that minimizes the sum of cover time of all the hyperedges. For all hyperedges of H , when $k_e = 1$, it boils down to the MSSC problem. Bansal et al. provide a 4.642 approximation algorithm for GMSSC, which is close to the approximation factor of 4 for MSSC [2]. For every $\epsilon > 0$, it is NP-hard to approximate MSSC within a ratio of $4 - \epsilon$ [12].

We now state some results from the literature regarding the hardness of approximability of MSVC and MSSC. It is NP-hard to approximate MSVC within a ratio better than $1 + \epsilon$ for some $\epsilon > 0$. For some $\rho < 4/3$ and every d , MSVC can be approximated within a ratio of ρ on d -regular graphs, but such a result doesn't hold in the case of MSSC. For every $\epsilon > 0$, there exists r, d such that it is NP-hard to approximate MSSC within a ratio better than $2 - \epsilon$ on r -uniform d -regular hypergraphs [12]. Recently, it was proved that under the assumption of Unique Games Conjecture, MSVC can not be approximated within 1.014 [32].

Although MSVC is NP-hard, there are some classes of graphs for which it is polynomial time solvable. Gera et al. provide the cost of MSVC in polynomial time for complete bipartite graphs, biregular bipartite graphs, multi-stars, hypercubes, prisms, etc. [19]. They also provide upper and lower bounds for the cost of MSVC in terms of independence number, the girth of a graph, and vertex cover number. MSVC is polynomial-time solvable for split graphs and caterpillars [31]. Interestingly, it is an open question whether MSVC for trees is polynomial-time solvable or NP-hard for a long time [31].

MSSC is studied in the realm of online algorithms, too. A constructive deterministic upper bound of $O(r^{3/2} \cdot n)$, where r is an upper bound for the cardinality of subsets, for online MSSC was given by Fotakis [15]. This bound was then improved to $O(r^4)$ by Bienkowski and Mucha [6]. Though this bound removed the dependency on n , this was existential. Basiak et al. gave a constructive and improved bound of $O(r^2)$ [5].

Our Methods and Results. We study MSVC from the perspective of parameterized complexity. A natural parameter to consider for the FPT algorithm is the solution size. But, for any connected graph G on n vertices, and an optimal ordering ϕ , $\mu_G(\phi) \geq n - 1$. Hence, MSVC parameterized by the solution size is trivially in FPT. Many NP-hard problems are generally tractable when parameterized by treewidth. However, vertex ordering problems are an exception to that. For MSVC, it is even harder to consider the treewidth as a parameter because we do not know if MSVC on trees is polynomial-time solvable or not.

The vertex cover number is a parameter used to prove the tractability of many vertex ordering problems. As MSVC aims to minimize the sum of the cover time of all the edges, it is natural to consider the vertex cover number as the parameter. For a vertex cover S of size k , we define a relation on $I = V \setminus S$ such that two vertices of I are related to each other if their neighborhood is the same. This is an equivalence relation that partitions I into equivalence classes. Interestingly, we prove that all vertices in the same equivalence class appear consecutively in an optimal ordering. So, we guess the relative ordering of vertices in S and then guess the locations of equivalence classes. This gives an FPT algorithm for MSVC parameterized by vertex cover.

MSVC is polynomial-time solvable for a few classes of graphs, with complete graphs being one among them. In fact, any ordering of vertices of K_n is an optimal ordering. A set $M \subset V(G)$ is called a *clique modulator* of the graph G , if the graph induced on $V(G) \setminus M$ is a clique. Finding an optimal solution for MSVC is non-trivial if G has a clique modulator of size k . Hence, we consider the size of a clique modulator as another parameter for MSVC. Vertex cover and clique modulator are complementary parameters. If a graph G has a vertex cover of size k , then the same set of vertices is a clique modulator for the complement of graph G . Hence, these two parameters fit well together in understanding the complete picture of tractability. Although clique modulator is complementary parameter of vertex cover, finding an optimal solution of MSVC with clique modulator is more challenging than that of vertex cover. We use similar approach by defining the same neighborhood relation on clique vertices. Unfortunately, all the vertices of an equivalence class need not be consecutive here in any optimal ordering. But, we prove that there is an optimal ordering σ with the following property. Let ℓ_1, \dots, ℓ_k be the location of the vertices from the modulator. Then, for any $i \in [k-1]$ and any equivalence class A , the vertices from A in the locations $\ell_i + 1, \dots, \ell_{i+1} - 1$ are consecutive in σ . Also, for any equivalence class A , the vertices from A in the locations $1, \dots, \ell_1 - 1$ as well as in the locations $\ell_k - 1, \dots, n$ are consecutive. We formulate an Integer Quadratic Programming for finding the number of vertices from each equivalence class present between two modulator vertices and provide an FPT algorithm. We summarize our results as follows:

- MSVC can be solved in $[k!(k+1)^{2^k} + (1.2738)^k]n^{O(1)}$ time, where k is the size of a minimum vertex cover.
- MSVC can be solved in $f(k) \cdot n^{O(1)}$ time for some computable function f , where k is the size of a minimum clique modulator.

Other Related Works. Given a graph $G(V, E)$ on n vertices and a bijection $\phi : V(G) \rightarrow [n]$, let a $cost(G, \phi)$ be defined on the vertex ordering ϕ of G . Vertex ordering problems are to minimize the $cost(G)$ over all possible orderings ϕ . Based on the definition of the cost function, there are various vertex ordering problems studied in the literature.

BANDWIDTH of a graph G , a vertex ordering problem, has been extensively studied since 1960. It has applications in speeding up many matrix computations of symmetric matrices. It is NP-complete [30], even for many restricted classes of graphs [16, 29]. Approximating BANDWIDTH with a constant factor is NP-hard even for trees [11], but it has an FPT approximation algorithm [10]. It is W[1]-hard parameterized by cluster vertex deletion number [20]. Though BANDWIDTH parameterized by its solution size is in XP [21], it is fixed-parameter tractable parameterized by vertex cover number [13], and neighborhood diversity [1].

CUTWIDTH of a graph G , often referred to as the Min-Cut Linear Arrangement problem in the literature, is NP-complete [18], even when restricted to graphs with maximum degree 3 [27]. However, it is polynomial-time solvable on trees [34]. For a given k , there is a linear time algorithm that outputs the cut if the CUTWIDTH of G is at most k , else outputs that the graph has CUTWIDTH more than k [33]. CUTWIDTH is FPT parameterized by vertex cover [13].

The OPTIMAL LINEAR ARRANGEMENTS (OLA) problem came up in the minimization of error-correcting codes and was first studied on n -dimensional cubes [23]. It is NP-complete [17], but polynomial-time solvable for trees [9]. OLA parameterized above-guaranteed value (guaranteed value: $|E(G)|$) [14, 22], vertex cover [26] is in FPT.

These vertex ordering problems are studied in the field of parameterized algorithms and approximation algorithms. MSVC is well-explored in the realm of approximation algorithms. But to the best of our knowledge, it is not studied in the realm of parameterized complexity.

2 Preliminaries and Notations

Let G be a graph with $V(G)$ as the set of vertices and $E(G)$ as the set of edges. Let σ be an ordering of $V(G)$ and $X \subseteq V(G)$. The ordering σ restricted on X is denoted by $\sigma|_X$. It represents the relative ordering between the vertices of X . A bijection $\sigma|_X : X \rightarrow \{1, 2, \dots, |X|\}$ is defined such that for all $u, v \in X$,

$$\sigma(u) < \sigma(v) \text{ if and only if } \sigma|_X(u) < \sigma|_X(v).$$

For an ordering ϕ , we represent $\phi(u) < \phi(v)$ as $u \prec_\phi v$ as well. We omit the subscript ϕ when the ordering is clear. Let X and Y be disjoint subsets of $V(G)$, such that in an ordering ϕ , $u \prec v$, for all $u \in X, v \in Y$. If the vertices of X are consecutive in ϕ ; likewise, the vertices of Y are also consecutive, though there could be vertices between X and Y , we represent this as $\phi(X) \prec \phi(Y)$.

Definition 2. *The right degree of a vertex v in an ordering ϕ is defined as $rd_\phi(v) = |\{u \in N_G(v) : \phi(u) > \phi(v)\}|$.*

Lemma 1 (\star^3). *In an optimal ordering ϕ , the sequence of right degrees of vertices is non-increasing.*

Lemma 2 (\star). *In an ordering ϕ of a graph G , locations of two consecutive non-adjacent vertices of equal right degrees can be swapped to get a new ordering with the same cost.*

The input to an Integer Quadratic Programming (IQP) problem is an $n \times n$ integer matrix Q , an $m \times n$ integer matrix A , and an m -dimensional integer vector b . The task is to find a vector $x \in \mathbb{Z}^+$ such that $x^T Q x$ is minimized subject to the constraints $Ax \leq b$. Let L be the total number of bits required to encode the input IQP.

Proposition 1 (Lokshtanov [26]). *There exists an algorithm that, given an instance of Integer Quadratic Programming, runs in time $f(n, \alpha)L^{O(1)}$, and determines whether the instance is infeasible, feasible and unbounded, or feasible and bounded. If the instance is feasible and bounded, the algorithm outputs an optimal solution. Here α is the largest absolute value of an entry of Q and A .*

³ Proofs of results marked with \star are omitted due to paucity of space.

3 MSVC Parameterized by Vertex Cover

Let G be a graph with the size of a minimum vertex cover of G at most k , where k is a positive integer. Placing the vertices of a minimum vertex cover first in the ordering appears to be an appealing approach to achieve MSVC. However, graph G_2 in Figure 1b shows this isn't always optimal. The red-colored vertices form the minimum vertex cover of G_2 . Consider an ordering ψ with the first eight locations having these vertices. The order among these vertices doesn't matter, as each vertex covers exactly two unique edges (each edge is covered by exactly one vertex of vertex cover). And the remaining vertices of the graph are located 9^{th} location onwards, in any order. The cost of such an ordering is 72. Consider an ordering ϕ , such that $\phi(v_1) = 1$, and v_1 is followed by $v_6, v_9, v_{12}, v_{15}, v_7, v_{10}, v_{13}, v_{16}$ respectively. The remaining vertices are ordered in any way after v_{16} . The cost of ϕ is 62. So, positioning the vertices of minimum vertex cover of size k , in the first k locations, doesn't always yield an optimal solution. Hence, a careful analysis is required for designing an FPT algorithm.

Let $S = \{v_1, v_2, \dots, v_k\}$ be a minimum vertex cover of graph G . Then, $I = V(G) \setminus S$ is an independent set. And $N_S(u) = \{v \in S \mid \{u, v\} \in E(G)\}$. There are total $k!$ relative orderings of vertices of S . Consider one such ordering, $v_{i_1} \prec v_{i_2} \prec \dots \prec v_{i_k}$, where i_1, i_2, \dots, i_k is a permutation of $[k]$. The set of vertices of I that appear before v_{i_1} is represented by *Block 1*. The set of vertices between v_{i_j} and $v_{i_{j+1}}$ is represented as *Block $j + 1$* , for $j \in [k - 1]$. And the set of vertices after v_{i_k} is called *Block $k + 1$* . This depiction is shown in Figure 2. In an optimal



Fig. 2: $k + 1$ blocks in an ordering of $V(G)$

ordering, a vertex from I can be in any one of the $k + 1$ blocks. Consider the relation R on I as follows: uRw if $N_S(u) = N_S(w)$. This equivalence relation R partitions I into at most 2^k equivalence classes.

Lemma 3 (\star). *There is an optimal ordering ϕ such that, in every block, all the vertices from the same equivalence class are consecutive.*

Lemma 4. *Let S be a minimum vertex cover of G . There is an optimal ordering such that for every equivalence class A , all the vertices from A are consecutive.*

Proof. To prove the lemma, it is enough to show that there is an optimal ordering σ such that the following conditions are satisfied:

- (i) Each equivalence class is contained in exactly one block.
- (ii) For every block, all the vertices from the same equivalence class are consecutive.

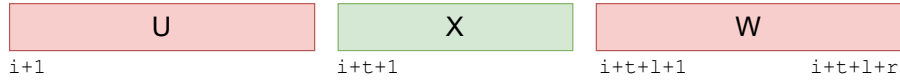
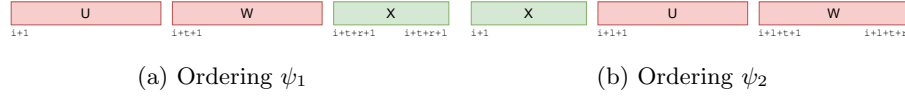

 Fig. 3: Vertices of equivalence class A distributed in different blocks


Fig. 4: Cases of Lemma 4

Given an ordering ϕ , let f_ϕ be a function defined on the equivalence classes as

$f_\phi(A)$: the number of blocks that contain vertices from equivalence class A .

Notice that $f_\phi(A) \geq 1$, for all A . Let $Q(\phi) = \sum_A f_\phi(A)$.

Any optimal ordering can be modified to satisfy (ii) by Lemma 3, with the cost being unaffected. Let α be the number of equivalence classes formed by the relation R on $I = V(G) \setminus S$. We only need to prove there is an optimal ordering σ , satisfying (ii), with $Q(\sigma) = \alpha$. For contradiction, assume in every optimal solution satisfying (ii), there is at least one equivalence class whose vertices are not entirely contained in a single block. Consider one with the least value of Q , say ϕ , among all the optimal orderings, satisfying (ii). At least one equivalence class A exists whose vertices are not in one block.

Let $U = \{u_1, u_2, \dots, u_t\}$ be a subset of A , with $\phi(u_1) = i + 1, \phi(u_2) = i + 2, \dots, \phi(u_t) = i + t$, where $t \geq 1$ and $i \geq 0$. And $\phi^{-1}(i + t + 1) \notin A$. Let $X = \{\phi^{-1}(i + t + 1), \dots, \phi^{-1}(i + t + \ell)\}$ be such that $X \cap S \neq \emptyset$. Let us denote these vertices in X as x_1, x_2, \dots, x_ℓ . Let $W = \{w_1, w_2, \dots, w_r\}$ be the first appearance of vertices from A after U , with $\phi(w_1) = i + t + \ell + 1, \phi(w_2) = i + t + \ell + 2, \dots, \phi(w_r) = i + t + \ell + r$, with $r \geq 1$, as shown in Figure 3.

Let v_1, v_2, \dots, v_p be the vertices from S , adjacent to A , such that $\phi(v_1) = i_1, \phi(v_2) = i_2, \dots, \phi(v_p) = i_p$, where $i + t < i_1 < i_2 < \dots < i_p \leq i + t + \ell$. There can be more vertices from S in X that are not adjacent to A , but we do not care about their location. Let the vertices in W have the right degree d . Then the vertices in U have the right degree $d + p$.

We claim that $p \geq 1$. If $p = 0$, then the right degree of vertices in U and W is d . And by Lemma 1, all the vertices between U and W have the right degree d . Hence, by repeated application of Lemma 2, U can be shifted to W or W to U without affecting the cost and minimizing the value of $Q(\phi)$, a contradiction.

Case(i): Consider a new ordering ψ_1 , where the entire set W is shifted next to U . The weight for all the edges incident on vertices up to U and after W , from ϕ will remain unchanged. The weight changes for edges incident on X and W . The set W shifts to the left by ℓ locations, and the set X shifts to the right by r locations, as shown in Figure 4a.

We analyze the change in the cost because of the shift. Consider a vertex $w \in W$, with right degree d and location a in ϕ . So, w contributes $a \cdot d$ in the cost of ϕ . Because of the shift, vertex w is shifted by exact ℓ locations towards the left. It now contributes $(a - \ell) \cdot d$ in the cost of ψ . The change in the cost because of w is $-\ell \cdot d$. This holds true for each vertex in W . As there are r vertices, each with the right degree d , the total cost change is $-\ell dr$.

The degrees of all the vertices of X , which are not adjacent to W , remain unchanged by this shift. Let x_i be one such vertex at location b . It contributes $b \cdot rd_\phi(x_i)$ in ϕ . Because of the shift of the r locations, it now contributes $(b + r) \cdot rd_\phi(x_i)$ in ψ . Hence, the change in the cost is $r \cdot rd_\phi(x_i)$. The increased cost due to all such vertices is

$$\left[\sum_{i \in [\ell] \setminus \{i_1, i_2, \dots, i_p\}} rd_\phi(x_i) \right] \cdot r.$$

A modulator vertex v_p is adjacent to all the vertices of W . But $w \prec_{\psi_1} v_p$. So, the right degree of v_p decreases by $|W| = r$. And its location shifts by r locations. As it is the same for all the vertices v_1, \dots, v_p , the change in the cost due to these vertices is

$$\left[\sum_{i \in \{i_1, \dots, i_p\}} (rd_\phi(x_i) - r) \right] \cdot r$$

The weight of an edge $e = \{v_1, w_1\}$ is i_1 in ϕ . But, $w_1 \prec_{\psi_1} v_1$, and e 's weight now is the location of w_1 in ψ_1 , i. e. $i + t + 1$. Hence, the change in the weight is $(i + t + 1 - i_1)$. Similarly, for the edge $\{v_1, w_j\}$, for $j \in [r]$, change in its weight is $(i + t + j - i_1)$. In the same manner, the change in the weight of the edges $\{v_2, w_j\}$ becomes $(i + t + j - i_2)$, for $j \in [r]$. And it follows a similar pattern for edges incident on v_3, \dots, v_p as well. The weight of the rest of the edges remains unaffected. Hence, the total change in the cost is: $\mu_G(\psi_1) - \mu_G(\phi) =$

$$\begin{aligned} & -\ell dr + \left[\sum_{i \notin \{i_1, i_2, \dots, i_p\}, i=1}^{\ell} rd_\phi(x_i) \right] \cdot r + \left[\sum_{i \in \{i_1, \dots, i_p\}} (rd_\phi(x_i) - r) \right] \cdot r \\ & + \sum_{j=1}^r (i + t + j - i_1) + \sum_{j=1}^r (i + t + j - i_2) + \dots + \sum_{j=1}^r (i + t + j - i_p) \\ & = -\ell dr + r \cdot \sum_{i=1}^{\ell} rd_\phi(x_i) - pr^2 - r(i_1 + \dots + i_p) + p[(i + t + 1) + \dots + (i + t + r)] \\ & = -\ell dr + r \cdot \sum_{i=1}^{\ell} rd_\phi(x_i) - pr^2 - r(i_1 + \dots + i_p) + pir + ptr + pr(r + 1)/2 \\ & = r \cdot \sum_{i=1}^{\ell} rd_\phi(x_i) - r \cdot y - pr^2/2 \quad (\text{where, } y = \ell d + (i_1 + \dots + i_p) - pi - pt - p/2) \end{aligned}$$

So, we have, $\mu_G(\psi_1) - \mu_G(\phi) = r \cdot \left[\sum_{i=1}^l rd_\phi(x_i) - y - pr/2 \right]$.

As $r > 0$, if $\sum_{i=1}^l rd_\phi(x_i) < y + pr/2$, then $\mu_G(\psi_1) - \mu_G(\phi) < 0$. (1)

Case(ii): Let ψ_2 be a new ordering obtained by swapping the positions of U and X . It is shown in Figure 4b. Weights of the edges $\{u_j, v_1\}$, $j \in [t]$, change from $i+j$ to i_1-t . Similarly, weights of the edges $\{u_j, v_q\}$, $j \in [t]$, $q \in [p]$, change from $i+j$ to i_q-t . For the remaining d edges incident on U , their weight increases by ℓ . As there are t vertices in U each of degree d , the total increase cost is ℓdt . And for the remaining edges of X , their weights are reduced by the factor t . The total change in the cost function is as follows: $\mu_G(\psi_2) - \mu_G(\phi) =$

$$\begin{aligned} & \ell dt - \left[\sum_{i=1}^{\ell} rd_\phi(x_i) \right] \cdot t + \sum_{j=1}^t [i_1 - t - (i + j)] \\ & + \sum_{j=1}^t [i_2 - t - (i + j)] + \cdots + \sum_{j=1}^t [i_p - t - (i + j)] \\ & = \ell dt - t \sum_{i=1}^{\ell} rd_\phi(x_i) - pt^2 + t(i_1 + \cdots + i_p) - p[(i+1) + (i+2) + \cdots + (i+t)] \\ & = -t \cdot \sum_{i=1}^{\ell} rd_\phi(x_i) + ty - pt^2/2 \quad (\text{as, } y = \ell d + (i_1 + \cdots + i_p) - pi - pt - p/2) \end{aligned}$$

We have $\mu_G(\psi_2) - \mu_G(\phi) = t \cdot \left[- \sum_{i=1}^{\ell} rd_\phi(x_i) + y - pt/2 \right]$.

As $t > 0$, we have, if $\sum_{i=1}^{\ell} rd_\phi(x_i) > y - pt/2$, then $\mu_G(\psi_2) - \mu_G(\phi) < 0$ (2)

Notice that $p, t, r > 0$, this implies $\sum_{i=1}^{\ell} rd_\phi(x_i) > y - pt/2$ or $\sum_{i=1}^{\ell} rd_\phi(x_i) < y + pr/2$ (or both). Hence, corresponding to the sum of the right degrees of vertices in X , shifting U towards W or vice versa gives an ordering with a cost less than the optimal value, a contradiction. Hence, our assumption that Q is greater than the number of equivalence classes in every optimal ordering is wrong. \square

Let $\mathcal{S} = \{\sigma_1, \sigma_2, \dots, \sigma_{k!}\}$ be the set of all permutations of vertices in S , and A_1, A_2, \dots, A_q be all the equivalence classes. By Lemma 4, each equivalence class is contained in exactly one block, and vertices from each equivalence class are consecutive. With $k+1$ blocks and q equivalence classes, there are $(k+1)^q$ possible choices. We denote a choice as c_j for $j \in \{1, 2, \dots, (k+1)^q\}$. Let \mathcal{E} denote the set of all possible choices. A configuration (σ_i, c_j) corresponds to the permutation σ_i of \mathcal{S} , and choice c_j of \mathcal{E} for the equivalence classes, where

$i \in [k!], j \in [(k+1)^q]$. An equivalence class, once fixed in one of the $k+1$ blocks, in a configuration, (σ_i, c_j) , its right degree can be easily calculated, as it is adjacent to only vertices from vertex cover. In a configuration (σ_i, c_j) , the relative order between the equivalence classes in any block is decided by their right degrees by Lemma 1. The equivalence classes are arranged such that their right degrees are non-increasing. A configuration with minimum cost is an optimal ordering for MSVC.

It takes $(1.2738)^k n^{O(1)}$ time to find a vertex cover of size at most k . Hence, the FPT running time of the algorithm is $[k!(k+1)^{2^k} + (1.2738)^k] n^{O(1)}$. Correctness follows from Lemma 4.

4 MSVC Parameterized by Clique Modulator

In this section, we prove that the MSVC parameterized by the clique modulator is fixed-parameter tractable.

Definition 3. *A set $M \subset V(G)$ is called a clique modulator of the graph G , if the graph induced on $V(G) \setminus M$ is a clique.*

The input of the problem is a positive integer k , and graph G with the size of a minimum clique modulator at most k . The question is to find an optimal ordering for MSVC of G . We use Integer Quadratic Programming (IQP) to construct the solution for MSVC parameterized by the clique modulator. IQP can be solved in time $f(t, \alpha)n^{O(1)}$ time, where f is some computable function, and t is the number of variables in IQP, and α is an upper bound for the coefficients of variables in IQP [26].

Let $M = \{v_1, v_2, \dots, v_k\}$ be a clique modulator of the graph G . Then, $Q = V(G) \setminus M$ is a clique. Consider the relation R on Q as follows: a vertex u is related to w if $N_M(u) = N_M(w)$. It is an equivalence relation, partitioning Q into equivalence classes A_1, A_2, \dots, A_ℓ , where $\ell \leq 2^k$.

In any optimal ordering, each equivalence class doesn't need to be entirely contained within one block, unlike when the parameter is a vertex cover. But, in an optimal solution, vertices from each equivalence class can be arranged consecutively within each block, with the cost being unaffected (see Lemma 6). Once the relative ordering of modulator vertices is guessed, Lemma 1 enables us to determine the ordering of equivalence classes in each block. The distinguishing factor to get this ordering is their neighbors in M . For a given relative ordering σ_M of M , for each block $B_i, i \in [k+1]$, we define right modulator degree for a vertex $u \in B_i$ as

$$\text{rm}_{\sigma_M}(u, i) = |\{v \in N_M(u) : \sigma_M(v) \geq i\}|.$$

For any two vertices u and v in an equivalence class A , $\text{rm}_{\sigma_M}(u, i) = \text{rm}_{\sigma_M}(v, i)$. We call this the right modulator degree of equivalence class A , $\text{rm}_{\sigma_M}(A, i)$ in B_i .

First, we prove that for any ordering σ of $V(G)$, if we sort the equivalence classes based on their right modulator degree in each block, then the cost of this new ordering is at most the cost of σ . Let $\sigma_M = \sigma|_M$.

Lemma 5 (*). *Let σ_M be a permutation of M . Let $g_1 : [\ell] \rightarrow \{A_1, A_2, \dots, A_\ell\}$ be a bijection such that $\text{rm}_{\sigma_M}(g_1(1), 1) \geq \text{rm}_{\sigma_M}(g_1(2), 1) \geq \dots \geq \text{rm}_{\sigma_M}(g_1(\ell), 1)$. For any ordering σ of $V(G)$, with $\sigma|_M = \sigma_M$, let $\hat{\sigma}$ be an ordering of $V(G)$ such that*

$$\hat{\sigma}(Y \cap g_1(1)) \prec \hat{\sigma}(Y \cap g_1(2)) \prec \dots \prec \hat{\sigma}(Y \cap g_1(\ell)) \prec \sigma_{>|Y|}$$

where the set of vertices of the clique $(V(G) \setminus M)$ in block 1 of σ is denoted by Y , and $\sigma_{>|Y|}$ denotes the ordering of vertices after the first $|Y|$ locations. Then, $\mu_G(\hat{\sigma}) \leq \mu_G(\sigma)$.

Now, we extend this analysis to the remaining blocks.

Lemma 6. *Let σ_M , a permutation of M , be given. Let $g_i : [\ell] \rightarrow \{A_1, A_2, \dots, A_\ell\}$ be a bijection such that $\text{rm}_{\sigma_M}(g_i(1), i) \geq \text{rm}_{\sigma_M}(g_i(2), i) \geq \dots \geq \text{rm}_{\sigma_M}(g_i(\ell), i)$, for all $i \in [k+1]$. For any ordering σ of $V(G)$, with $\sigma|_M = \sigma_M$, let $\hat{\sigma}$ be an ordering of $V(G)$ such that*

$$\hat{\sigma} = \hat{\sigma}_1(Y_1) \prec \sigma_M^{-1}(1) \prec \hat{\sigma}_2(Y_2) \prec \dots \prec \sigma_M^{-1}(k) \prec \hat{\sigma}_{k+1}(Y_{k+1}).$$

where the set of vertices of the clique $(V(G) \setminus M)$ in block i of σ is denoted by Y_i and for each $i \in [k+1]$, $\hat{\sigma}_i(Y_i) = (Y_i \cap g_i(1)) \prec (Y_i \cap g_i(2)) \prec \dots \prec (Y_i \cap g_i(\ell))$. Then, $\mu_G(\hat{\sigma}) \leq \mu_G(\sigma)$.

Proof. Repeatedly apply the technique of Lemma 5 for each block $i \in [k+1]$, to get the ordering $\hat{\sigma}$. The proof follows from Lemma 5. \square

We call a permutation of the form $\hat{\sigma}$ in Lemma 6, a nice permutation.

4.1 Integer Quadratic Programming

Let G be the input graph and M be a clique modulator of size k . And $Q = V(G) \setminus M$ is a clique of size $n - k$. Let A_1, \dots, A_ℓ be the number of equivalence classes that partition Q .

As a first step, we guess the ordering σ_M of M such that there is an optimum ordering σ with $\sigma|_M = \sigma_M$. Lemma 6 implies that we know the ordering of vertices within each block irrespective of the number of vertices from equivalence classes in each block. We use IQP to determine the number of vertices from each equivalence class that will be present in different blocks. Thus, for each $i \in [\ell]$ and $j \in [k+1]$, we use a variable x_{ij} that represents the number of vertices from A_i in the block j .

Recall that Q is a clique of size $n - k$. Consider an arbitrary ordering of vertices of Q . Its cost, denoted by $\mu(Q)$ is

$$\mu(Q) = 1 \cdot (n - k - 1) + 2 \cdot (n - k - 2) + \dots + (n - k - 1)(1).$$

This cost remains consistent for any ordering of Q .

For any ordering ϕ of $V(G)$, $\mu_G(\phi) \geq \mu(Q)$. And $\mu(Q)$ can be found beforehand. We treat $\mu(Q)$ as the base cost for G . We need to minimize the cost after

introducing the modulator vertices. Let the first modulator vertex v_1 be introduced in an ordering of Q . Assume, there are n_1 vertices of Q after v_1 . Then, the weight of each edge whose both endpoints are after v_1 increases by 1. There are $\binom{n_1}{2}$ such clique edges. Hence, the total increase in the cost with respect to the clique edges is $\binom{n_1}{2}$. A modulator vertex v_i is introduced after v_{i-1} and $v_{i-1} \prec v_i$, for all $i \in \{2, \dots, k\}$. Let n_i be the number of clique vertices after v_i in the ordering. Hence, inclusion of v_i causes the cost to increase by $\binom{n_i}{2}$. The total increase in the weights of clique edges after including all the modulator vertices is $\sum_{i=1}^k \binom{n_i}{2}$.

Thus, in the IQP, we also use variables n_1, n_2, \dots, n_k . To minimize the cost, we need to find the ordering of modulator vertices to be introduced and their location. Since σ_M is fixed, we know the relative ordering of equivalence classes in each block using Lemma 6. Recall that we have a variable x_{ij} for each equivalence class A_i and block B_j . The variable n_i represents the number of vertices from Q after v_i in the hypothetical optimum solution σ . This implies that in IQP, we need to satisfy the constraints of the following form.

$$\begin{aligned} n_p &= \sum_{j=p+1}^{k+1} \sum_{i=1}^{\ell} x_{ij} && \text{for all } p \in [k] \\ |A_i| &= \sum_{j=1}^{k+1} x_{ij} && \text{for all } i \in [\ell] \end{aligned}$$

We explained that after introducing the modulator vertices, the increase in cost due to edges in the clique is given by the expression $\sum_{i=1}^k \binom{n_i}{2}$. Next, we need an expression regarding the increase in cost due to edges incident on modulator vertices. Towards that, we introduce a variable y_p for each p to indicate the location of the modulator vertex v_p in the hypothetical optimal ordering σ . Clearly, we should satisfy the following constraints.

$$y_p = n - (n_p + k - p) \quad \text{for all } p \in [k]$$

There are n_p vertices from Q after v_p and $k - p$ vertices from M after v_p in the hypothetical optimum ordering σ . Now we explain how to get the increase in the cost due to edges incident on the modulator vertex v_p . We know its position is y_p . We need to find its right degree. We use a variable d_p to denote the right degree of p . Let $\text{rm}_p = |\{v \in N_M(v_p) : \sigma_M(v) > \sigma_M(v_p)\}|$. That is, rm_p is the number of edges between the modulator vertices with one endpoint v_p and the other in $\{v_{p+1}, \dots, v_k\}$. Let $I_p \subseteq [\ell]$ such that $i \in I_p$ if and only if v_p is adjacent to the vertices in A_i . Then, the number of neighbors of v_p in the clique Q that appear to the right of v_p in the hypothetical solution is $\sum_{j=p+1}^{k+1} \sum_{i \in I_p} x_{ij}$. This implies that we have the following constraints.

$$d_p = \text{rm}_p + \sum_{j=p+1}^{k+1} \sum_{i \in I_p} x_{ij} \quad \text{for all } p \in [k]$$

The increase in cost due to edges incident on modulator vertices such that the endpoints of those edges in the modulator appear before the other endpoints is $\sum_{p=1}^k d_p \cdot y_p$. Now, we need to consider the cost of edges between the clique vertices and the modulator vertices such that clique vertices appear before the modulator vertex. We use r_{ij} to denote the right modulator degree of vertices in the equivalence class A_i from j^{th} block. It is defined as follows. Let $u \in A_i$ be a fixed vertex. Then, $r_{ij} = |\{v \in N_M(u) : \sigma_M(v) \geq j\}|$. Notice that r_{ij} is a constant, less than or equal to k . Let y_{ij} denote the location of the first vertex from the equivalence class A_i in block j in the hypothetical solution σ . Then, we should have the following constraint. Here, we use Lemma 6 and recall the bijection g_j . Let $J_{i,j} \subseteq [\ell]$ be such that $q \in J_{i,j}$ if and only if $g_j^{-1}(A_q) < g_j^{-1}(A_i)$.

$$y_{ij} = y_{j-1} + \sum_{q \in J_{i,j}} x_{qj}.$$

Here, we set $y_0 = 0$. The cost due to edges from A_i in block j to modulator vertices to its right in the ordering is

$$r_{ij}[y_{ij} + (y_{ij} + 1) + \dots + (y_{ij} + x_{ij} - 1)] = r_{ij} \left(x_{ij} \cdot y_{ij} + \binom{x_{ij}}{2} \right).$$

Thus, we summarize our IQP as follows.

Minimize $\sum_{p=1}^k d_p \cdot y_p + \sum_{i=1}^k \binom{n_i}{2} + \sum_{i=1}^{\ell} \sum_{j=1}^{k+1} r_{ij} (x_{ij} \cdot y_{ij} + \binom{x_{ij}}{2})$
Subject to

$$\begin{aligned} n_p &= \sum_{j=p+1}^{k+1} \sum_{i=1}^{\ell} x_{ij} && \text{for all } p \in [k] \\ |A_i| &= \sum_{j=1}^{k+1} x_{ij} && \text{for all } i \in [\ell] \\ y_p &= n - (n_p + k - p) && \text{for all } p \in [k] \\ d_p &= \text{rm}_p + \sum_{j=p+1}^{k+1} \sum_{i \in I_p} x_{ij} && \text{for all } p \in [k] \\ y_{ij} &= y_{j-1} + \sum_{q \in J_{i,j}} x_{qj} && \text{for all } i \in [\ell] \text{ and } j \in [k+1] \end{aligned}$$

Here, n, k, r_{ij} and rm_p are constants. Also, $r_{ij}, \text{rm}_p \leq k$ for all i, j and p . This completes the construction of an IQP instance for a fixed permutation σ_M of M . Notice that, there are at most $2^k(k+1)$ variables x_{ij} and $2^k(k+1)$ variables y_{ij} . And there are k variables for y_p and n_p each. Hence, the total number of variables in the above IQP formulation is at most $2^{k+1} \cdot (k+1) + 2k$.

Next, we prove that the coefficients in the objective functions and the constraints are upper bounded by a function of k . The coefficient of $y_p \cdot d_p$ is at most

1 in the objective function. The coefficients of n_i and n^2 are also upper bounded by a constant. As $r_{ij} \leq k$, the coefficients of $(x_{ij})^2$, $x_{ij} \cdot y_{ij}$, and x_{ij} are all at most k . Similarly, coefficients are bounded above by $O(k)$ in the constraints. Thus, by Proposition 1, we can solve the above IQP instance in time $f(k)n^{O(1)}$ time for a computable function f .

Lemma 7 (\star). *Let σ_M be an ordering of M such that there is an optimum ordering σ with $\sigma|_M = \sigma_M$. An optimal solution to the IQP defined above is an optimal solution to MSVC, and vice versa.*

For each ordering σ_M of M , we construct an IQP instance as described above and solve. Finally, we consider the best solution among them and construct an ordering from it according to the explanation given for the IQP formulation. Correctness of the algorithm follows from Lemma 7.

References

1. Bakken, O.R.: Arrangement problems parameterized by neighbourhood diversity. Master’s thesis, The University of Bergen (2018)
2. Bansal, N., Batra, J., Farhadi, M., Tetali, P.: Improved approximations for min sum vertex cover and generalized min sum set cover. In: Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms (SODA). pp. 998–1005 (2021)
3. Bar-Noy, A., Bellare, M., Halldórsson, M.M., Shachnai, H., Tamir, T.: On chromatic sums and distributed resource allocation. *Information and Computation* **140**(2), 183–202 (1998)
4. Barenholz, U., Feige, U., Peleg, D., et al.: Improved approximation for min-sum vertex cover. <http://wisdomarchive.wisdom.weizmann.ac.il> **81**, 06–07 (2006)
5. Basiak, M., Bienkowski, M., Tatarczuk, A.: An improved deterministic algorithm for the online min-sum set cover problem. arXiv preprint arXiv:2306.17755 (2023)
6. Bienkowski, M., Mucha, M.: An improved algorithm for online min-sum set cover. In: AAAI. pp. 6815–6822 (2023)
7. Burer, S., Monteiro, R.D.: A projected gradient algorithm for solving the maxcut sdp relaxation. *Optimization methods and Software* **15**(3-4), 175–200 (2001)
8. Chen, J., Kanj, I.A., Xia, G.: Improved upper bounds for vertex cover. *Theoretical Computer Science* **411**(40-42), 3736–3756 (2010)
9. Chung, F.R.: On optimal linear arrangements of trees. *Computers & mathematics with applications* **10**(1), 43–60 (1984)
10. Dregi, M.S., Lokshtanov, D.: Parameterized complexity of bandwidth on trees. In: ICALP. pp. 405–416. Springer (2014)
11. Dubey, C., Feige, U., Unger, W.: Hardness results for approximating the bandwidth. *Journal of Computer and System Sciences* **77**(1), 62–90 (2011)
12. Feige, U., Lovász, L., Tetali, P.: Approximating min sum set cover. *Algorithmica* **40**, 219–234 (2004)
13. Fellows, M.R., Lokshtanov, D., Misra, N., Rosamond, F.A., Saurabh, S.: Graph layout problems parameterized by vertex cover. In: Algorithms and Computation: 19th International Symposium, ISAAC. pp. 294–305. Springer (2008)
14. Fernau, H.: Parameterized algorithmics for linear arrangement problems. *Discrete Applied Mathematics* **156**(17), 3166–3177 (2008)

15. Fotakis, D., Kavouras, L., Koumoutsos, G., Skoulakis, S., Vardas, M.: The on-line min-sum set cover problem. In: 47th International Colloquium on Automata, Languages, and Programming, ICALP. pp. 51:1–51:16 (2020)
16. Garey, M.R., Graham, R.L., Johnson, D.S., Knuth, D.E.: Complexity results for bandwidth minimization. *SIAM Journal on Applied Mathematics* **34**(3), 477–495 (1978)
17. Garey, M.R., Johnson, D.S., Stockmeyer, L.: Some simplified np-complete problems. In: Symposium on the Theory of Computing (STOC). pp. 47–63 (1974)
18. Gavril, F.: Some np-complete problems on graphs. In: Proc. Conf. on Inform. Sci. and Systems, 1977. pp. 91–95 (1977)
19. Gera, R., Rasmussen, C., Stanica, P., Horton, S.: Results on the min-sum vertex cover problem. Tech. rep., NAVAL POSTGRADUATE SCHOOL MONTEREY CA DEPT OF APPLIED MATHEMATICS (2006)
20. Gima, T., Kim, E.J., Köhler, N., Melissinos, N., Vasilakis, M.: Bandwidth parameterized by cluster vertex deletion number. arXiv preprint arXiv:2309.17204 (2023)
21. Gurari, E.M., Sudborough, I.H.: Improved dynamic programming algorithms for bandwidth minimization and the mincut linear arrangement problem. *Journal of Algorithms* **5**(4), 531–546 (1984)
22. Gutin, G., Rafey, A., Szeider, S., Yeo, A.: The linear arrangement problem parameterized above guaranteed value. *Theory of Computing Systems* **41**, 521–538 (2007)
23. Harper, L.H.: Optimal assignments of numbers to vertices. *Journal of the Society for Industrial and Applied Mathematics* **12**(1), 131–135 (1964)
24. Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of computer computations Proc. Sympos., pp. 85–103 (1972)
25. Khot, S., Regev, O.: Vertex cover might be hard to approximate to within $2 - \epsilon$. *Journal of Computer and System Sciences* **74**(3), 335–349 (2008)
26. Lokshantov, D.: Parameterized integer quadratic programming: Variables and coefficients. arXiv preprint arXiv:1511.00310 (2015)
27. Makedon, F.S., Papadimitriou, C.H., Sudborough, I.H.: Topological bandwidth. *SIAM Journal on Algebraic Discrete Methods* **6**(3), 418–444 (1985)
28. Mohan, S.R., Acharya, B., Acharya, M.: A sufficiency condition for graphs to admit greedy algorithm in solving the minimum sum vertex cover problem. In: International Conference on Process Automation, Control and Computing. pp. 1–5. IEEE (2011)
29. Monien, B.: The bandwidth minimization problem for caterpillars with hair length 3 is np-complete. *SIAM Journal on Algebraic Discrete Methods* **7**(4), 505–512 (1986)
30. Papadimitriou, C.H.: The np-completeness of the bandwidth minimization problem. *Computing* **16**(3), 263–270 (1976)
31. Rasmussen, C.W.: On efficient construction of minimum-sum vertex covers (2006)
32. Stanković, A.: Some Results on Approximability of Minimum Sum Vertex Cover. In: Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2022). vol. 245, pp. 50:1–50:16 (2022)
33. Thilikos, D.M., Serna, M., Bodlaender, H.L.: Cutwidth i : A linear time fixed parameter algorithm. *Journal of Algorithms* **56**(1), 1–24 (2005)
34. Yannakakis, M.: A polynomial algorithm for the min-cut linear arrangement of trees. *Journal of the ACM (JACM)* **32**(4), 950–988 (1985)