Improved Approximation for Min-Sum Vertex Cover

Uri Barenholz [*] Uriel Feige [†] David Peleg [‡]

September 20, 2006

Abstract

The paper describes an approximation algorithm for the Min Sum Vertex Cover (MSVC) problem, achieving a constant approximation factor strictly smaller than 2, thus improving on the best currently known approximation algorithm for the problem.

[*]Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: `uri.barenholz@gmail.com`.

[†]Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel, and Microsoft Research, Redmond, Washington. E-mail: `uriel.feige@weizmann.ac.il`.

[‡]Department of Computer Science and Applied Mathematics, The Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: `david.peleg@weizmann.ac.il`. Supported in part by a grant from the Israel Science Foundation.

1 Introduction

Background: The *minimum sum vertex cover* (MSVC) problem is a special case of the *minimum sum set cover* problem. It is related both to the classical *minimum vertex cover* and to linear arrangement problems. The input to the problem is a graph, and the output is a linear ordering of the vertices of the graph. The time a vertex is covered is defined as its ordinal number in the ordering of the vertices, and the time an edge is covered is defined as the minimal time in which one of its vertices (a.k.a. endpoints) is covered. The objective is to order the vertices is such a way that the average time an edge is covered is minimal.

Linear arrangement problems on graphs often come up as heuristics for speeding up matrix computations. Indeed, the MSVC problem came up in [2, Section 4] in the context of designing efficient algorithms for solving semidefinite programs. This was the motivation for the study of the problem.

Related work: In [2], the MSVC problem is applied as a tool for speeding up solvers of semidefinite programs. It is approximated by a greedy algorithm, picking the vertices in descending order according to their degree. A similar greedy algorithm is analyzed in [1] in the context of *min sum coloring*, and is shown to approximate it within a factor of 4, given the independent sets of the conflict graph.

A set of related linear arrangement problems is presented in [4], and an analysis of different approaches for solving them is given. In particular, *min sum set cover* (MSSC) is presented as a generalization of MSVC. It is shown that the analysis of [1] for the greedy algorithm can be applied to MSSC and MSVC as well, yielding an approximation ratio of 4 on both. Furthermore, it is shown that for every $\varepsilon > 0$, it is NP-hard to approximate MSSC within a ratio of $4-\varepsilon$. For MSVC, a linear-programming based algorithm using a randomized rounding technique is presented, achieving an approximation ratio bounded by 2. It is shown that there is some (unknown) constant such that it is NP-hard to approximate the problem within a better factor. For d-regular graphs it is shown that MSVC can be approximated within a ratio of ρ for some $\rho < 4/3$, and every d.

Although MSSC and MSVC are closely related to set cover and vertex cover problems (such as *min set cover* and *max k-cover*, for which tight approximation bounds are known, and more), there is a major difference between them. MSVC (and hence MSSC) is not "linear", in the sense that given an instance of MSVC which is composed of two disjoint instances, the optimal solution is not necessarily a combination of the optimal solutions to each of the sub-instances. An example (given in [4]) is the following. Let G_1 be a graph on 9 vertices $u, v_1, w_1, \ldots, v_4, w_4$, in which vertex u is connected by a star to vertices $v_1, \ldots, v_4$, and for every $1 \leq i \leq 4$, v_i is connected to w_i. The optimal solution to MSVC first uses u to cover 4 edges, and then covers the remaining edges one by one. However, if G is the disjoint union of G_1 and G_2, where G_2 is a graph consisting of three isolated edges, then the optimal solution for MSVC becomes to first take $v_1, \ldots v_4$, and then cover the remaining edges of G_2 one by one. As noted in [4], this difference makes it more difficult to design and analyze algorithms for MSVC.

Our contributions: We revisit the linear programming relaxation that was previously used in [4] for MSVC. We show how to modify the rounding technique that was presented in [4] so as to obtain an approximation ratio strictly better than 2. Specifically, we show an approximation ratio of 1.99995, though our analysis is not tight and a better approximation bound is probably achievable.

Our results (as well as the factor 2 approximation previously given in [4]) extend to a weighted version of MSVC, in which edges have nonnegative weights and the goal is to find a linear order minimizing the weighted average time by which an edge is covered. Details about this extension are omitted from this manuscript since they are fairly standard: as is often the case, the approximation ratio achieved by linear programming relaxations is not sensitive to nonnegative weights in the objective function.

The remainder of the paper is structured as follows. In Section 2 we review the algorithm described in [4] and its analysis. This presentation is used as a basis for the definition of a modified rounding technique, described in Section 3. The modified rounding technique is analyzed in Section 4, and it is shown that for every instance, either the original rounding technique or the modified one yield a constant approximation ratio strictly smaller than 2. Section 5 concludes with some suggestions for further study.

2 Ratio 2 approximation for MSVC

This section reviews the algorithm of [4], named RR herein. This algorithm is later used as a subprocedure in our algorithm. We also detail the analysis of algorithm RR, since the analysis of the new algorithm builds upon it.

2.1 The linear program

Given a graph G on n vertices, indexed as $\{1, \ldots, n\}$, consider the following linear program Π. The program makes use of variables x_{it} (for $1 \leq i \leq n$ and $t \geq 1$) and y_{ijt} (for (i, j) an edge in G, to be denoted as $(i, j) \in G$, and for $t \geq 1$).

Program Π: Minimize $\Pi(G) = \sum_{(i,j)\in G} \sum_t y_{ijt}$ subject to

(C1) $\sum_i x_{it} \leq 1$, for every $t \geq 1$.

(C2) $y_{ijt} \geq 1 - \sum_{t'<t}(x_{it'} + x_{jt'})$, for every $(i,j) \in G$ and $t \geq 1$.

(C3) $0 \leq x_{it} \leq 1$, for all x_{it}.

(C4) $0 \leq y_{ijt} \leq 1$, for all y_{ijt}.

By defining $X_{kt} = \displaystyle\sum_{t'<t} x_{kt}$, constraint (C2) can be rewritten as

(C2A) $y_{ijt} \geq 1 - (X_{it} + X_{jt})$.

We convert Π to an integer program Π′ (with variables x'_{ij} and y'_{ijt} respectively) by converting constraints (C3) and (C4) to integrality constraints, i.e., $x'_{it} \in \{0,1\}$ and $y'_{ijt} \in \{0,1\}$. Note that in Π′ the variable x'_{it} is an indicator variable indicating whether vertex i is chosen at step t, and y'_{ijt} is an indicator variable indicating whether edge (i,j) is still uncovered before step t. Therefore, Π′ is a representation of MSVC as an integer program, implying that an optimal solution to Π′ is a solution to MSVC.

The linear program Π is solvable in polynomial time. The resulting fractional solution provides a lower bound for the optimal solution for MSVC.

Let us record some basic properties of Π for later use.

Lemma 2.1 *Let $\langle x, y\rangle$ be an optimal solution for Π on a graph G. Then for every $(i,j) \in G$ and $t \geq 1$ the following holds.*

1. $y_{ijt} = \max(1 - (X_{it} + X_{jt}), 0)$.
2. *If* $y_{ij(t+1)} > 0$*, then* $y_{ijt} = y_{ij(t+1)} + x_{it} + x_{jt}$.

2.2 The randomized rounding algorithm RR

Given a fractional solution $\langle x, y\rangle$ to the linear program, the following randomized rounding technique RR is applied to obtain a rounded solution. Note that in the fractional solution, X_{it} represents the cumulative contribution of node i to the coverage of the edge (i,j).

For each vertex i, let t_i be the largest value of t for which $X_{it} < 1/2$. (If this holds for every t, set $t_i = \infty$.) Introduce new variables z_{it}, defined as follows:

$$z_{it} = \begin{cases} 2x_{it} & \text{, for } \quad t < t_i, \\ 1 - \sum_{t'<t_i} z_{it'} & \text{, for } \quad t = t_i, \\ 0 & \text{, for } \quad t > t_i. \end{cases}$$

Note that always

$$z_{it} \leq 2x_{it}. \tag{1}$$

Also note that $z_{it} \geq 0$ for every i and t, and that $\sum_t z_{it} = 1$. Hence, for a fixed i, the vector $(z_{it})_{t \geq 1}$ can be viewed as a probability vector.

Randomly select the timestep t at which vertex i is chosen by using the probability vector z_i (i.e., taking z_{it} as the probability that vertex i is chosen at timestep t). For the chosen t, set $\check{x}_{it} = 1$ and for all other timesteps t', set $\check{x}_{it'} = 0$. Note that the resulting rounded solution $\check{x}_{it}$ is a cover since in an optimal solution to the linear program Π every edge (i,j) is covered at some point t in time (i.e., $y_{ijt} = 0$), implying by constraint (C2A) that either $X_{it} \geq 1/2$ or $X_{jt} \geq 1/2$ (or both).

Also note that the rounded solution might violate the first constraint, (C1), since the timestep selection is done independently for each vertex, implying that more than one vertex can be chosen at a given timestep. Hereafter, we refer to this intermediate rounded solution as the *Timestep phase*, and the term *timestep* will always be used to denote a partition of time into distinct steps without restriction on the number of vertices chosen at each such step.

For every timestep t, denote the number of vertices chosen at timestep t by $s_t = \sum_i \check{x}_{it}$. Replace *timestep* t by s_t *timeslots* and assign the vertices originally chosen at *timestep* t to these new *timeslots*, one vertex per timeslot, at a random order. The variables $\hat{x}_{it}$ will be used to denote the new vertex selection sequence. The resulting solution $\hat{x}_{it}$ again satisfies the first constraint (C1). Hereafter, the term *timeslots* will always be used to denote a sub-partition of the *timesteps* into slots, at each of which exactly one vertex (out of the vertices originally chosen at the corresponding timestep) is chosen.

Using the rounded solution $\hat{x}_{it}$ for the selection of vertices, assign to $\hat{y}_{ijt}$ the optimal values possible. Output the solution $\langle \hat{x}, \hat{y} \rangle$, namely, the values $\{\hat{x}_{it}\}$ and $\{\hat{y}_{ijt}\}$.

2.3 Analysis of the integrality gap

For a given input graph G, denote by $\mathsf{RR}(G)$ the expected value of the solution selected by procedure RR on G. It obviously holds that $\Pi(G) \leq \Pi'(G) \leq \mathsf{RR}(G)$. Let

$$W_{ij} = \sum_{0 \leq t < \infty} y_{ijt} \tag{2}$$

be the cost of the edge (i,j) when applying Π to G. Note that $\Pi(G) = \sum_{(i,j) \in G} W_{ij}$.

Let $\widehat{W_{ij}}$ denote the expectation of the timestep at which the edge (i,j) is covered when applying RR to G. Denote by p_{ijt} the probability that the edge (i,j) is not covered by algorithm

RR prior to timestep t. The random variable w_{ijt} (defined for every edge $(i, j) \in G$ and every timestep t) denotes the number of vertex selections in timestep t prior to the cover of (i, j). In particular, if neither i nor j gets selected during timestep t, then w_{ijt} is the total number of timeslots timestep t is split into. Note that w_{ijt} is defined conditioned on (i, j) not being covered prior to timestep t. Then it holds that

$$\widehat{W_{ij}} = \sum_{0 \le t < \infty} p_{ijt} \mathbb{E}(w_{ijt}) \qquad \text{and} \qquad \mathsf{RR}(G) = \sum_{(i,j) \in G} \widehat{W_{ij}} \ . \tag{3}$$

Before beginning the analysis, we state the following helpful lemma.

Lemma 2.2 *Let* $0 \le a, b \le 1/2$. *Then* $a + b \ge 4ab$.

We now analyze the approximation ratio of RR.

Lemma 2.3 *If* $t \le \min(t_i, t_j)$ *then*

$$\begin{aligned} p_{ijt} &= 1 - 2(X_{it} + X_{jt}) + 4X_{it}X_{jt} \\ &= y_{ijt} - (X_{it} + X_{jt}) + 4X_{it}X_{jt}. \end{aligned}$$

Proof First note that the probability p_{ijt} satisfies

$$p_{ijt} = \left(1 - \sum_{t' < t} z_{it'}\right)\left(1 - \sum_{t' < t} z_{jt'}\right).$$

As $t \le \min(t_i, t_j)$, by the definition of z_{it} we can substitute $2x_{it}$ for z_{it} and the same for j, and get

$$p_{ijt} = \left(1 - \sum_{t' < t} 2x_{it'}\right)\left(1 - \sum_{t' < t} 2x_{jt'}\right) = (1 - 2X_{it})(1 - 2X_{jt}) = 1 - 2(X_{it} + X_{jt}) + 4X_{it}X_{jt},$$

and by Lemma 2.1 (1)

$$p_{ijt} = y_{ijt} - (X_{it} + X_{jt}) + 4X_{it}X_{jt}. \qquad \blacksquare$$

We use this lemma to get the following.

Lemma 2.4 $p_{ijt} \le y_{ijt}$.

Proof If $t > t_i$, then $X_{it} \ge 1/2$ and hence $p_{ijt} = 0 \le y_{ijt}$ (and similarly for $t > t_j$). Otherwise, $t \le \min(t_i, t_j)$. In this case, by Lemma 2.3,

$$p_{ijt} = y_{ijt} - (X_{it} + X_{jt}) + 4X_{it}X_{jt}.$$

As $X_{it} \leq 1/2$ and $X_{jt} \leq 1/2$ we get (using Lemma 2.2)

$$X_{it} + X_{jt} \;\geq\; 4X_{it}X_{jt}$$

and therefore $p_{ijt} \leq y_{ijt}$. ∎

Lemma 2.5 $\mathbb{E}(w_{ijt}) \leq 2$.

Proof Recall that the definition of w_{ijt} is conditioned on the edge (i,j) not being covered prior to timestep t. Due to this condition it cannot be assumed that the total expected number of vertices selected at timestep t is 2. Without the conditioning, and using the inequality in Equation (1) as well as constraint (C1), it holds that the expected value is simply

$$\sum_k z_{kt} \;\leq\; 2\sum_k x_{kt} \;\leq\; 2.$$

Let r_t (r standing for "rest") denote the number of vertices chosen at timestep t excluding i and j. The expectation of r_t can be calculated (using Equation (1) and constraint (C1) again) as

$$\mathbb{E}(r_t) \;=\; \sum_{k \neq i,j} z_{kt} \;\leq\; 2\sum_{k \neq i,j} x_{kt} \;\leq\; 2.$$

The proof analyzes three different cases.

1. In case neither x_{it} nor x_{jt} are rounded to 1 in timestep t, $\mathbb{E}(w_{ijt}) = \mathbb{E}(r_t) \leq 2$.
2. In case either x_{it} or x_{jt} are rounded to 1 in timestep t, but not both, the expected number of timeslots within timestep t is $1 + \mathbb{E}(r_t) \leq 3$. Therefore the expected waiting time $\mathbb{E}(w_{ijt})$ is the expected number of vertices preceding the selected vertex in the sequence (of size at most 3), plus one. Since the location is random, the expected waiting time is $\mathbb{E}(w_{ijt}) = \mathbb{E}(r_t)/2 + 1 \leq 2$.
3. In case both x_{it} and x_{jt} are rounded to 1 in timestep t, the expected number of timeslots within timestep t is $2 + \mathbb{E}(r_t) \leq 4$. Therefore the expected waiting time $\mathbb{E}(w_{ijt})$ is the expected number of vertices preceding the selection of the first out of i or j in the sequence (of size 4), plus one. This is calculated as $\mathbb{E}(w_{ijt}) = \mathbb{E}(r_t)/3 + 1 \leq 5/3 < 2$.

To conclude, in all of these cases $\mathbb{E}(w_{ijt}) \leq 2$, proving the lemma. ∎

Corollary 2.6 *For every graph G and for every edge (i,j), it holds that*

$$\widehat{W}_{ij} \;\leq\; 2W_{ij}, \quad \textit{implying that} \;\; \mathsf{RR}(G) \;\leq\; 2\Pi(G).$$

Proof Using Lemmas 2.4 and 2.5 we get that $p_{ijt} \cdot \mathbb{E}(w_{ijt}) \leq 2y_{ijt}$. Therefore, for every edge (i, j),

$$\widehat{W_{ij}} = \sum_{0 \leq t < \infty} p_{ijt} \mathbb{E}(w_{ijt}) \leq 2 \sum_{0 \leq t < \infty} y_{ijt} = 2W_{ij}.$$

Summing over all edges $(i, j) \in G$ yields

$$\mathsf{RR}(G) = \sum_{(i,j) \in G} \widehat{W_{ij}} \leq 2 \sum_{(i,j) \in G} W_{ij} = 2\Pi(G). \qquad \blacksquare$$

Note that as the fractional solution lower bounds the optimal integral one, this yields a bound over the approximation ratio for MSVC.

RR can be made deterministic using the method of conditional expectation, thus achieving an approximation ratio as good as the expected one in polynomial time. This is done iteratively in the following way. Mark all vertices as "undecided". Pick an undecided vertex, k. For every timestep t for which $z_{it} > 0$, check the expectation of the total value of $\Pi(G)$ conditioned on k being selected at t, and all the other "decided" vertices selected according to their assigned selection timesteps. As the unconditional expectation (i.e., when k's selection timestep is undecided) is the sum of these conditional expectations with weights according to z_{kt}, at least one of the conditional expectation terms must be smaller than or equal to the unconditional expectation. Denote the timestep for which this holds by t_k. Assign the selection timestep of k to be t_k, and mark k as "decided". Repeat until all vertices are "decided".

The split of the timesteps to timeslots can be derandomized in a similar manner.

3 A modified rounding algorithm MRR

3.1 The reorganization step

We modify RR as follows. After the timestep phase, perform the following *reorganization* step:

- For every timestep t, and for every vertex i selected at that timestep, change the vertex timestep selection to $t - \epsilon_1$ with probability p, or to $t + \epsilon_2$ with probability $1 - p$ (where p, ϵ_1 and ϵ_2 are parameters depending on t).

We refer to the modified algorithm as MRR.

To illustrate the potential gains achievable by applying the reorganization step over RR, let us furnish the following simplified analysis of w_{ijt}. Recall that (assuming neither i nor j are selected prior to timestep t) there are three possibilities:

- Neither i nor j are selected in timestep t.
- Exactly one (either i or j) is selected in timestep t.
- Both i and j are selected during timestep t.

For the sake of the simplified analysis, we fix $p = 1/2$ and $\epsilon_1 = \epsilon_2 = 1$, and analyze the effect of the reorganization step in each of the three possibilities. We ignore (for the time being) effects caused due to boundary conditions (in particular, for $t = 1$ the reorganization step is not yet defined).

In case neither i nor j are selected during timestep t, $\mathbb{E}(w_{ijt})$ is unchanged, yielding $\mathbb{E}(w_{ijt}) = \mathbb{E}(r_t) = 2$. This holds since all the vertices rounded to 1 in timestep t, prior to the reorganization step, will move either one timestep forward or one timestep backward, with equal probability. On the other hand (assuming that $\mathbb{E}(r_{t-1}) = \mathbb{E}(r_{t+1}) = 2$), in expectation, one vertex from timestep $t-1$ and one vertex from timestep $t+1$ will move to timestep t. Note that for the sake of simplicity we neglect the case that either i, j, or both are rounded to 1 in timestep $t+1$ and one of them moves to timestep t in the reorganization step. Such a case only reduces the expectation. Therefore, the expected number of vertices rounded to 1 in timestep t remains unchanged following the reorganization step.

In case either i or j (but not both) is rounded to 1 during timestep t, assume w.l.o.g. that i is the vertex rounded to 1 in timestep t. With probability 1/2 it moves to the previous timestep, resulting in $\mathbb{E}(w_{ijt}) = 0$ for this timestep. With probability 1/2 it moves to the next timestep resulting in a total waiting time of $\mathbb{E}(w_{ijt}) = \mathbb{E}(r_t) + \mathbb{E}(w_{ij(t+1)}) = 2 + \mathbb{E}(w_{ij(t+1)}) \leq 4$, where $\mathbb{E}(r_t) = 2$ due to 2 vertices moving into timestep t, and $\mathbb{E}(w_{ij(t+1)}) \leq 2$ due to the expected waiting time during timestep $t+1$. Again, for simplicity we neglect the case that j is rounded to 1 in timestep $t+1$ and moves into timestep t. As noted before, this case can only reduce the expectation. The total expected waiting time in this case is therefore still bounded by 2.

In case both i and j are rounded to 1 during timestep t there is a 3/4 probability that at least one of them will move to timestep $t-1$, resulting in $\mathbb{E}(w_{ijt}) = 0$, and a probability of 1/4 that both will move to timestep $t+1$, resulting in $\mathbb{E}(w_{ijt}) = \mathbb{E}(r_t) + \mathbb{E}(w_{ij(t+1)}) \leq 4$. The total expected waiting time is therefore reduced to 1 in this case!

We conclude that in the third case we gain by applying the reorganization step.

3.2 An addition to the original algorithm

A slight modification is needed for RR to make the analysis of MRR complete. At step 2, after defining z_{it}, add a new set of variables defined as follows:

$$z'_{it} = \begin{cases} 0 & \text{, for} \quad t < t_i, \\ \left(2\sum_{t' \le t_i} x_{it'}\right) - 1 & \text{, for} \quad t = t_i, \\ 2x_{it} & \text{, for} \quad t > t_i. \end{cases}$$

To ensure that $(z'_{it})_{t \ge 1}$ also forms a probability vector for every i, we define $z'_{i\infty} = 1 - \sum_{t<\infty} z'_{it}$. We extend RR to select a *second* instance of every vertex i using the new probability vector, $(z'_{it})_{t \ge 1}$. Note that this modification does not affect the analysis of the integrality gap of RR, and the approximation ratio of RR after this modification remains 2. However, it does imply that the expected number of vertices (or instances of vertices) covering each edge (i, j) that are selected during the execution of RR is at least 2. This claim is used in the upcoming analysis.

3.3 A parametrized version of MRR

We now define a parametrized version of MRR, using a parameter $0 < \mu < 1$, which will be used to determine the values of ϵ_1 and ϵ_2. The parametrized version of the algorithm will be denoted by MRR(μ). In particular, we restrict our selection by assuming that the time shifts ϵ_1 and ϵ_2 are linear functions of t, and that $p = 1/2$. We can therefore redefine the reorganization step as follows:

- For every timestep t, and for every vertex i selected at that timestep, change the vertex timestep selection to $t - \epsilon_1(\mu)$ with probability $1/2$, or to $t + \epsilon_2(\mu)$ with probability $1/2$.

Denoting $\epsilon_1(\mu) = \mu_1 t$ and $\epsilon_2(\mu) = \mu_2 t$, we require that $0 < \mu_1, \mu_2 < 1$.

A problem that can arise in the parametrized version of the algorithm is due to "pushing" too many vertices to a few timesteps. This occurs if $\mu_1 >> \mu_2$, in which case many vertices are expected to "jump" to an earlier timestep, or stay more or less at the same timestep. Obviously this incurs a penalty applying to all edges covered later on, because after performing the reorganization step each of the first timesteps will include more vertices than it did originally. Each such timestep will therefore be divided into a larger number of timeslots. In order to avoid this problem we require that, on the average, the number of vertices in each timestep remains unchanged. This is achieved by analyzing "cuts", and ensuring the following property:

Property 3.1 *For each timestep t, the number of vertices that jump (on average) from timesteps smaller than t to timesteps larger than or equal to t is equal to the number of vertices that jump from timesteps larger than or equal to t to timesteps smaller than t.*

The average number of vertices that jump from timesteps smaller than t to timesteps larger than or equal to t is equal to the number of timesteps in times $t' < t$ for which $t' + \mu_2 t' \geq t$ (assuming an average of two vertices per timestep and a probability of 1/2 for jumping to a later timestep). In the other direction, the number of vertices that jump from timesteps larger than or equal to t to timesteps smaller than t is equal to the number of timesteps for which $t' \geq t$ and $t' - \mu_1 t' < t$. The number of timesteps corresponding to the first condition is exactly $t - t_-$ where $t_- + \mu_2 t_- = t$, implying $t_- = t/(1+\mu_2)$. Similarly the number of timesteps corresponding to the second condition is exactly $t_+ - t$ where $t_+ - \mu_1 t_+ = t$, implying $t_+ = t/(1-\mu_1)$. The difference is therefore

$$(t - t_-) - (t_+ - t) \; = \; t\left(2 - \frac{1}{1-\mu_1} - \frac{1}{1+\mu_2}\right) \; \equiv \; Dt.$$

If $D > 0$, the difference will grow with time, meaning the density will decrease. If $D < 0$, the density will increase. We require that $D = 0$ (thus looking for the smallest difference between μ_1 and μ_2 that still doesn't cause penalty due to increase in the density). This is achieved by requiring that

$$2 \; = \; \frac{1}{1-\mu_1} + \frac{1}{1+\mu_2}, \qquad \text{or,} \qquad \mu_2 \; = \; \frac{\mu_1}{1-2\mu_1}.$$

We therefore define

$$\epsilon_1(\mu) \; = \; \mu_1 t \; = \; \mu t \quad \text{and} \quad \epsilon_2(\mu) \; = \; \mu_2 t \; = \; \frac{\mu}{1-2\mu} t.$$

3.4 A motivating analysis for MRR(μ)

In a similar manner to the simplified analysis of Section 3.1, the following possibilities exist for every edge (i, j):

If only i or j are selected at the timestep t at which (i, j) is first covered, then the expected timestep at which (i, j) is covered after the reorganization step is at most

$$(t - \mu_1 t)/2 + (t + \mu_2 t)/2 \; = \; t + \mu\mu_2 t,$$

where the equality is due to the definition of μ_1 and μ_2. This is an upper bound since if i and j are selected at relatively close timesteps, it is possible that the one chosen at the later timestep will move to a timestep smaller than that the one chosen earlier ends up in. We neglect this effect for the time being.

If both i and j are selected at the same timestep t then the expected cover time of edge (i, j) changes to

$$\frac{3}{4}(t - \mu_1 t) + \frac{1}{4}(t + \mu_2 t) \; = \; t + \frac{\mu_2(3\mu - 1)}{2} t,$$

again using the definition of μ_1 and μ_2. In this case, if $\mu_1 < 1/3$ then the added factor is negative, implying an improvement in the covering time of the edge.

Looking at the worst example analyzed in [4], which is composed of a large set of disjoint edges, each of which is covered by assigning a value of 1/2 to each of its endpoints at some timestep t, the second case always occurs. In this case we therefore expect (neglecting boundary conditions for the time being) an improvement of the approximation ratio by a factor of $\delta = \frac{\mu_2(3\mu_1-1)}{2}$. Selecting, for example, $\mu_1 = \mu = 1/5$ and hence $\mu_2 = 1/3$, will yield an improvement by a factor of $\delta = 1/15$ over the original 2 factor, giving a total approximation ratio of $2 - \delta \cdot 2 = 28/15$ for that example.

3.5 Boundary condition effect for the first time step

The definition of the reorganization step did not specify how it works on the first timestep (where no preceding timesteps exist), in case a vertex should move to a previous timestep. In this case, vertices that should move to previous timesteps are left in the same timestep.

This creates a penalty as it violates the requirement that, on average, the number of vertices in each timestep remains unchanged (as for the first timestep, vertices might move to it from itself and following timesteps, but none moves from timesteps preceding it). We note that due to our selection of μ_1 and μ_2, the boundary condition penalty is constant. Assuming (as indeed is the case) that $\mu < 1$, the penalty is bounded (on average) by 1. However, this penalty applies to all the edges, meaning that the reorganization step adds (on average) a one step delay for the cover time of every edge.

Even though one can generate arbitrarily large graphs for which this delay is not negligible, we claim that for these graphs other techniques can be applied to find a good approximating solution. Specifically, for a given edge (i, j), if the delay is not negligible (i.e., the delay causes a more than ε change to the cost of that edge) then it implies that the edge's cost is at most $1/\varepsilon$, meaning it is covered by timestep $1/\varepsilon$. Therefore, by solving MSVC using an exhaustive method for the first $1/\varepsilon$ vertices (and applying MRR to the rest), we ensure that the effect is negligible.

4 Achieving a better approximation bound

4.1 Edge-wise analysis

Our analysis concentrates on the notion of the cost of an edge, or the contribution of an edge to the total cost of a solution to Π, as defined in Equation (2). Recall that $\widehat{W_{ij}}$ is the expected time step at which the edge (i, j) is covered when applying RR to G, as defined in Equation (3). We define $\overline{W_{ij}}$ as the expected time step at which the edge is covered when applying MRR to G. In a similar manner to (3) it then holds that

$$\mathsf{MRR}(G) \;=\; \sum_{(i,j)\in G} \overline{W_{ij}}.$$

We define the ratios $\widehat{R_{ij}} = \widehat{W_{ij}}/W_{ij}$ and $\overline{R_{ij}} = \overline{W_{ij}}/W_{ij}$. We show that there exist constants ε_1 and ε_2 such that for any input graph G and a fractional solution $\langle x, y\rangle$ (as $\Pi(G)$ may have more than one optimal assignment, the construction depends upon the specific assignment $\Pi(G)$ outputs), the edges of G can be divided into two groups, $E = E_a \cup E_b$, such that

1. the edges of E_a satisfy $\widehat{R_{ij}} \leq 2 - \varepsilon_1$ and $\overline{R_{ij}} \leq 2 + \frac{2\mu^2}{1-2\mu}$, and
2. the edges of E_b satisfy $\overline{R_{ij}} \leq 2 - \varepsilon_2$ and $\widehat{R_{ij}} \leq 2$.

Subsequently, for a given fractional solution to a given instance of MSVC, we can check the relative cost each group contributes to the total value of the fractional solution. According to the ratio between the contributions of the two groups, we apply one of the two rounding techniques, to yield an approximation ratio strictly better than 2.

4.2 Refined analysis of RR

We analyze two conditions under which RR yields an approximation ratio smaller than 2. We combine these conditions to define E_a.

Lemma 4.1 *For a given parameter* $0 < \alpha < 1$, *if* $y_{ijt} \leq 1 - \alpha$ *then* $p_{ijt} \leq (1 - \alpha)y_{ijt}$.

Proof If $t > t_i$, then $X_{it} \geq 1/2$ and hence $p_{ijt} = 0 \leq (1 - \alpha)y_{ijt}$ (and similarly for $t > t_j$) regardless of the value of y_{ijt}. Otherwise, $t \leq \min(t_i, t_j)$. In this case, by Lemma 2.3, we get that in order to prove $p_{ijt} \leq (1 - \alpha)y_{ijt}$, it suffices to show that

$$(1 - \alpha)(1 - (X_{it} + X_{jt})) \;\geq\; (1 - (X_{it} + X_{jt})) - X_{it} - X_{jt} + 4X_{it}X_{jt},$$

or, simplifying, that

$$(1 + \alpha)(X_{it} + X_{jt}) \;\geq\; \alpha + 4X_{it}X_{jt}.$$

Assuming that $X_{it} + X_{jt} = c$ for some constant c, we note that the right hand side of the inequality is maximal when $X_{it} = X_{jt} = c/2$. Therefore, the required inequality will hold for any X_{it} and X_{jt} satisfying that $(1+\alpha)c \geq \alpha + c^2$, or, $(c-1)(c-\alpha) \leq 0$. This holds wherever $\alpha \leq c \leq 1$. The latter condition holds by the premise of the lemma. ▮

For simplicity of notation define $\gamma = \sqrt{1-4\alpha}$ and $\psi(\alpha) = \frac{1-\gamma}{2}$. Note that assuming $0 < \alpha < 1/4$,

$$\alpha \; = \; \frac{1-\sqrt{1-4\alpha+4\alpha^2}}{2} \; < \; \psi(\alpha).$$

Lemma 4.2 *For a given parameter $0 < \alpha < 1/4$, if $\psi(\alpha) \le y_{ijt} \le 1-\psi(\alpha)$, then $p_{ijt} \le y_{ijt}-\alpha$.*

Proof If $t > t_i$, then $X_{it} \ge 1/2$ and hence $p_{ijt} = 0$ (and similarly for $t > t_j$). As $y_{ijt} \ge \psi(\alpha) > \alpha$, the lemma holds in this case.

Otherwise, $t \le \min(t_i, t_j)$. In this case, by Lemma 2.3, to prove the requirement that $p_{ijt} \le y_{ijt} - \alpha$ it suffices to show that

$$X_{it} + X_{jt} \; \ge \; \alpha + 4X_{it}X_{jt}.$$

As observed in the proof of Lemma 4.1, assuming that $X_{it} + X_{jt} = c$ for some constant c, the right hand side of the inequality is maximal when $X_{it} = X_{jt} = c/2$. The required condition thus transforms into the quadratic equation $c^2 - c + \alpha \le 0$, which yields the requirement

$$\psi(\alpha) \; \le \; c \; \le \; 1-\psi(\alpha).$$

Therefore, the required condition holds for every timestep t for which $\psi(\alpha) \le y_{ijt} \le 1-\psi(\alpha)$ (taking the boundaries induced on y_{ijt} from the boundaries calculated for c). The lemma follows. ▮

For a given fractional (optimal) assignment to the edge (i, j), defined by assignments to $(x_{it})_{0 \le t < \infty}$, $(x_{jt})_{0 \le t < \infty}$ and $(y_{ijt})_{0 \le t < \infty}$, we define the following.

- Let t_1 denote the first timestep for which $y_{ijt_1} \le 1-\psi(\alpha)$.
- Let t_2 denote the first timestep for which $y_{ijt_2} \le \psi(\alpha)$.
- Let t_3 denote the first timestep for which $y_{ijt_3} = 0$ (i.e., the time when the edge is fully covered).

These times turn out to be the natural break points for categorizing the edges for which $\widehat{R_{ij}} < 2$. Note that formally, t_1, t_2 and t_3 are functions of the edge (i, j) and should thus be denoted accordingly; we omit the reference to (i, j) for notational simplicity, and throughout we make sure it is clear from the context.

Let $Y_{ij}[t', t''] = \sum_{t' \le t < t''} y_{ijt}$. Similarly, let $P_{ij}[t', t''] = \sum_{t' \le t < t''} p_{ijt}$. For abbreviation, let $Y_{ij} = Y_{ij}[t_1, \infty]$, and similarly for P_{ij}.

With these definitions, we have the following corollaries.

Corollary 4.3 *For a given parameter $0 < \beta < 1$, if $Y_{ij}[t_1, t_3] \geq \beta W_{ij}$, then $\widehat{R_{ij}} \leq 2 - 2\alpha\beta$.*

Proof Starting with Equation (3), observe that by Lemma 2.5

$$\widehat{W_{ij}} = \sum_{0 \leq t < \infty} p_{ijt}\mathbb{E}(w_{ijt}) \leq 2P_{ij} = 2\left(P_{ij}[0, t_1] + P_{ij}[t_1, t_3]\right).$$

Using Lemmas 2.4 and 4.1 we get that

$$\widehat{W_{ij}} \leq 2\left(Y_{ij}[0, t_1] + (1 - \alpha)Y_{ij}[t_1, t_3]\right) = 2W_{ij} - 2\alpha Y_{ij}[t_1, t_3] \leq (2 - 2\alpha\beta)W_{ij}.$$

The lemma follows. ▮

Corollary 4.4 *For a given parameter $0 < \beta < 1$, if $t_2 - t_1 \geq \beta t_2$ then $\widehat{R_{ij}} \leq 2 - 2\alpha\beta$.*

Proof As in the previous proof, by Equation (3) and Lemma 2.5

$$\widehat{W_{ij}} = \sum_{0 \leq t < \infty} p_{ijt}\mathbb{E}(w_{ijt}) \leq 2P_{ij} = 2\left(P_{ij}[0, t_1] + P_{ij}[t_1, t_2] + P_{ij}[t_2, t_3]\right).$$

Using Lemmas 2.4, 4.1 and 4.2 we get that

$$\begin{aligned} \widehat{W_{ij}} &\leq 2\left(Y_{ij}[0, t_1] + \sum_{t_1 \leq t < t_2} (y_{ijt} - \alpha) + (1 - \alpha)Y_{ij}[t_2, t_3]\right) \\ &= 2\left(Y_{ij}[0, t_2] - \alpha(t_2 - t_1) + (1 - \alpha)Y_{ij}[t_2, t_3]\right). \end{aligned}$$

As $t_2 - t_1 \geq \beta t_2$, and since $y_{ijt} \leq 1$, implying that $t_2 \geq Y_{ij}[0, t_2]$, we deduce that $\alpha(t_2 - t_1) \geq \alpha\beta Y_{ij}[0, t_2]$. Substituting in the previous inequality yields

$$\begin{aligned} \widehat{W_{ij}} &\leq 2\left((1 - \alpha\beta)Y_{ij}[0, t_2] + (1 - \alpha)Y_{ij}[t_2, t_3]\right) \\ &\leq 2\left(W_{ij} - \alpha\beta Y_{ij}[0, t_2] - \alpha Y_{ij}[t_2, t_3]\right) \leq (2 - 2\alpha\beta)W_{ij}. \end{aligned}$$

The lemma follows. ▮

4.3 A global bound on the approximation ratio of MRR

Denote by ε_{ijt} the event that the edge (i, j) is first covered by algorithm RR in timestep t. To bound the performance of MRR(μ) on all edges we use the following lemma.

Lemma 4.5 *For every edge $(i, j) \in G$, $\overline{R_{ij}} \leq 2 + \frac{2\mu^2}{1-2\mu}$.*

Proof The expected time step in which the edge (i,j) is covered when applying algorithm RR can be calculated as

$$\widehat{W_{ij}} = 2 \sum_{1 \le t < \infty} \mathbb{P}(\varepsilon_{ijt}) t,$$

where the 2 factor stems from the fact that, as implied by the proof of Lemma 2.5, on average, every timestep preceding timestep t is split into two timeslots.

Using Corollary 2.6 and the definition of algorithm $\mathsf{MRR}(\mu)$, we deduce that the expected time step at which the algorithm covers the edge can be bounded by

$$\overline{W_{ij}} \le (1 + (\mu_2 - \mu_1)/2)\widehat{W_{ij}} \le 2(1 + (\mu_2 - \mu_1)/2) W_{ij}.$$

This holds since in case RR covered the edge at timestep t, $\mathsf{MRR}(\mu)$ will cover it in timestep $t - \mu_1 t$ with probability at least $1/2$ (this is only a lower bound since in case both of the edge's covering instances happened in timestep t, this probability is $3/4$) and in timestep $t + \mu_2 t$ with probability at most $1/2$. By substituting the values for μ_1 and μ_2, the lemma follows. ∎

4.4 Edge classification

Using Corollaries 4.3 and 4.4 as well as Lemma 4.5, we define the sets E_a and E_b (depending on the parameters α and β) as follows:

$$\begin{aligned}
E_a^1 &= \{(i,j) \in G \mid (1-\beta)t_2 \ge t_1\}, \\
E_a^2 &= \{(i,j) \in G \mid Y_{ij}[t_1, t_3] \ge \beta W_{ij}\}, \\
E_a &= E_a^1 \cup E_a^2, \\
E_b &= G \backslash E_a = \{(i,j) \in G \mid (1-\beta)t_2 < t_1 \text{ and } Y_{ij}[t_1, t_3] < \beta W_{ij}\}.
\end{aligned}$$

Substituting $\varepsilon_1 = 2\alpha\beta$, we note that E_a conforms to the assertions in Section 4.1:

Lemma 4.6 *If $(i,j) \in E_a$ then $\widehat{R_{ij}} \le 2 - \varepsilon_1$ and $\overline{R_{ij}} \le 2 + \frac{2\mu^2}{1-2\mu}$.*

Proof Assuming $(i,j) \in E_a$,

- If $(i,j) \in E_a^1$, then $t_2 - t_1 \ge \beta t_2$, and using Corollary 4.4 we get that $\widehat{R_{ij}} \le 2 - 2\alpha\beta$.
- If $(i,j) \in E_a^2$, then $Y_{ij}[t_1, t_3] \ge \beta W_{ij}$, and using Corollary 4.3 we get that $\widehat{R_{ij}} \le 2 - 2\alpha\beta$.

The bound $\overline{R_{ij}} \le 2 + \frac{2\mu^2}{1-2\mu}$ follows directly from Lemma 4.5. ∎

We proceed to prove that edges $(i,j) \in E_b$ satisfy $\overline{R_{ij}} \le 2 - \varepsilon_2$ for some positive ε_2.

4.5 Analysis of $\mathsf{MRR}(\mu)$ on edges of E_b

Subclassification of E_b edges: We evaluate the expected time step at which $\mathsf{MRR}(\mu)$ covers an edge under different conditions. Note that (referring to the timestep phase) a vertex k is chosen in timestep t with probabilities z_{kt} and z'_{kt}, meaning that if both are non-zero then k can be chosen up to two times in timestep t. Recall that according to the definitions of z_{kt} and z'_{kt}, it holds that if $t < t_k$ then $z'_{kt} = 0$, and that if $t \geq t_k$ then surely vertex k was chosen prior to timestep t.

We partition E_b into two subsets:

1. $E_b^{small} = \{(i,j) \in E_b | t_2 < \min(t_i, t_j)\}$, namely, the edges such that the number of instances an endpoint of the edge is chosen prior to timestep t_2 is 0, 1 or 2.
2. $E_b^{large} = \{(i,j) \in E_b | t_2 \geq \min(t_i, t_j)\}$, namely, the edges such that the number of instances an endpoint of the edge is chosen prior to timestep t_2 is 1, 2, 3 or 4. Edges of E_b^{large} are therefore surely covered by timestep t_2 in the timestep phase.

In the upcoming analysis we use the following notation.

- Let $\varphi_{m,t}$ denote the event that by timestep t, exactly m instances of an endpoint of the edge (i,j) were chosen in the timestep phase. We use φ_m as a shorthand for φ_{m,t_2}.
- Let ρ_{kt} denote the event that vertex k was chosen by timestep t. We use ρ_k as a shorthand for ρ_{kt_2}.
- Let $E(m,t)$ denote the expectation of the timestep at which the edge is covered due to the reorganization step, given that $\varphi_{m,t}$ occurred. Note that $E(m,t)$ is upper bounded by the expectation of the timestep at which the edge is covered due to the reorganization step, given that it was covered by m instances *at* timestep t. In our calculations we use the latter as an upper bound. We use E_m as a shorthand for $E(m, t_2)$.
- $\bar{E}$ is the expected number of timeslots each timestep is split into after the reorganization step. As the reorganization step has Property 3.1, Lemma 2.5 implies that $\bar{E} \leq 2$.

We note that

$$\begin{aligned}
E(1,t) &\leq \tfrac{1}{2}(t + t\mu_2) + \tfrac{1}{2}(t - t\mu_1) &&= t + \tfrac{1}{2}t(\mu_2 - \mu_1), \\
E_1 &\leq \tfrac{1}{2}(t_2 + t_2\mu_2) + \tfrac{1}{2}(t_2 - t_2\mu_1) &&= t_2 + \tfrac{1}{2}t_2(\mu_2 - \mu_1), \\
E_2 &\leq \tfrac{1}{4}(t_2 + t_2\mu_2) + \tfrac{3}{4}(t_2 - t_2\mu_1) &&= t_2 + \tfrac{1}{4}t_2\mu_2 - \tfrac{3}{4}t_2\mu_1, \\
E_3 &\leq \tfrac{1}{8}(t_2 + t_2\mu_2) + \tfrac{7}{8}(t_2 - t_2\mu_1) &&= t_2 + \tfrac{1}{8}t_2\mu_2 - \tfrac{7}{8}t_2\mu_1.
\end{aligned}$$

We recall the definition of ε_{ijt} as the event that the edge (i,j) is first covered by Algorithm RR in timestep t. We need the following lemma in our analysis.

Lemma 4.7 *If $y_{ijt} \le 1/4$ then $\mathbb{P}(\varepsilon_{ijt}) \le y_{ijt} - y_{ij(t+1)}$.*

Proof Note that ε_{ijt} can be described as the event that (i, j) is covered by Algorithm RR prior to timestep $t + 1$ but not prior to timestep t. Thus

$$\mathbb{P}(\varepsilon_{ijt}) \;=\; p_{ijt} - p_{ij(t+1)}.$$

Also note that if $t > \min(t_i, t_j)$ then surely the edge is covered by timestep t, implying that $\mathbb{P}(\varepsilon_{ijt}) = 0 \le y_{ijt} - y_{ij(t+1)}$ and proving the lemma in this case.

Otherwise, $t \le \min(t_i, t_j)$. We therefore use Lemma 2.3 and the definition of X_{kt} to get that

$$\begin{aligned}
\mathbb{P}(\varepsilon_{ijt}) &= 2(X_{i(t+1)} + X_{j(t+1)}) - 2(X_{it} + X_{jt}) + 4X_{it}X_{jt} - 4X_{i(t+1)}X_{j(t+1)} \\
&= 2(x_{it} + x_{jt}) + 4X_{it}X_{jt} - 4(X_{it} + x_{it})(X_{jt} + x_{jt}) \\
&= 2(x_{it} + x_{jt}) - 4(X_{it}x_{jt} + X_{jt}x_{it} + x_{jt}x_{it}) \\
&\le 2(x_{it} + x_{jt}) - 4\min(X_{it}, X_{jt})(x_{jt} + x_{it}) - 4x_{jt}x_{it},
\end{aligned}$$

and using Lemma 2.1 (2),

$$\begin{aligned}
\mathbb{P}(\varepsilon_{ijt}) &\le (2 - 4\min(X_{it}, X_{jt}))(y_{ijt} - y_{ij(t+1)}) - 4x_{jt}x_{it} \\
&\le (2 - 4\min(X_{it}, X_{jt}))(y_{ijt} - y_{ij(t+1)}).
\end{aligned}$$

By the premise of the lemma and by Lemma 2.1 (1), $1/4 \ge y_{ijt} = 1 - (X_{it} + X_{jt})$, or, $X_{ij} + X_{jt} \ge 3/4$. As $t \le \min(t_i, t_j)$, $\max(X_{it}, X_{jt}) \le 1/2$, implying that $\min(X_{it}, X_{jt}) \ge 1/4$. Therefore, for such timesteps, it holds that

$$\mathbb{P}(\varepsilon_{ijt}) \;\le\; (2 - 4\min(X_{it}, X_{jt}))(y_{ijt} - y_{ij(t+1)}) \;\le\; y_{ijt} - y_{ij(t+1)}. \qquad \blacksquare$$

Bounding $\overline{W_{ij}}$ for E_b^{small} edges: We assume that $\alpha \le 3/16$, implying $\psi(\alpha) \le 1/4$. We show the following.

Lemma 4.8 *If $(i, j) \in E_b^{small}$, then*

$$\overline{W_{ij}} \;\le\; 2t_2 + t_2\mu(2\mu_2 - (1 - 2\psi(\alpha))(1 + \mu_2)) + 2(1 + \mu\mu_2)t_2\psi(\alpha) + 2(1 + \mu\mu_2)Y_{ij}[t_2, t_3].$$

Proof For E_b^{small} edges, $\overline{W_{ij}}$ can be evaluated as the sum of the following terms.

- The probability that (i, j) is covered by one instance of its endpoints by timestep t_2, $\mathbb{P}(\varphi_1)$, times the expected timestep it will be covered at due to the reorganization step, given that φ_1 occurred, E_1, times the expected number of timeslots each timestep preceding it is split into, $\bar{E}$.

- The probability that (i,j) is covered by two instances of its endpoints by timestep t_2, $\mathbb{P}(\varphi_2)$, times the expected timestep it will be covered at due to the reorganization step, given that φ_2 occurred, E_2, times the expected number of timeslots each timestep preceding it is split into, $\bar{E}$.
- For any timestep $t > t_2$, the probability that (i,j) is first covered at timestep t, $\mathbb{P}(\varepsilon_{ijt})$, times the expected timestep it will be covered at due to the reorganization step given that ε_{ijt} occurred, which is upper bounded by $E(1,t)$, times the expected number of timeslots each timestep preceding it is split into, $\bar{E}$.

We therefore get that for these edges,

$$\begin{aligned}\overline{W_{ij}} &\leq \mathbb{P}(\varphi_1) \cdot E_1 \cdot \bar{E} \\ &+ \mathbb{P}(\varphi_2) \cdot E_2 \cdot \bar{E} \\ &+ \sum_{t_2 < t \leq t_3} (\mathbb{P}(\varepsilon_{ijt}) \cdot E(1,t) \cdot \bar{E}).\end{aligned}$$

We bound each of these terms separately.

We observe that under the assumption that $\max(X_{it_2}, X_{jt_2}) < 1/2$ it holds that

$$\begin{aligned}\mathbb{P}(\varphi_{1,t}) &= \mathbb{P}((\rho_{it} \cap \overline{\rho_{jt}}) \cup (\rho_{jt} \cap \overline{\rho_{it}})) &&= \mathbb{P}(\rho_{it})(1 - \mathbb{P}(\rho_{jt})) + \mathbb{P}(\rho_{jt})(1 - \mathbb{P}(\rho_{it})), \\ \mathbb{P}(\varphi_{2,t}) &= \mathbb{P}(\rho_{it} \cap \rho_{jt}) &&= \mathbb{P}(\rho_{it})\mathbb{P}(\rho_{jt}).\end{aligned}$$

Putting the above together, we get that $\overline{W_{ij}}$ is bounded by

$$\begin{aligned}\overline{W_{ij}} \leq \quad & 2t_2(\mathbb{P}(\rho_i)(1 - \mathbb{P}(\rho_j)) + \mathbb{P}(\rho_j)(1 - \mathbb{P}(\rho_i)))(1 + (\mu_2 - \mu_1)/2) \\ & +2t_2\mathbb{P}(\rho_i)\mathbb{P}(\rho_j)\left(1 + \left(\frac{1}{4}\mu_2 - \frac{3}{4}\mu_1\right)\right) \\ & +2\sum_{t_2 < t \leq t_3} t\mathbb{P}(\varepsilon ijt)(1 + (\mu_2 - \mu_1)/2) \\ = \quad & 2t_2(\mathbb{P}(\rho_i) + \mathbb{P}(\rho_j) - \mathbb{P}(\rho_i)\mathbb{P}(\rho_j)) \\ & +t_2\left(\mathbb{P}(\rho_i) + \mathbb{P}(\rho_j) - \frac{3}{2}\mathbb{P}(\rho_i)\mathbb{P}(\rho_j)\right)(\mu_2 - \mu_1) \\ & -t_2\mathbb{P}(\rho_i)\mathbb{P}(\rho_j)\mu_1 \\ & +2\sum_{t_2 < t \leq t_3} t\mathbb{P}(\varepsilon_{ijt})(1 + (\mu_2 - \mu_1)/2).\end{aligned}$$

The worst assignment (in terms of the cost $\mathsf{MRR}(\mu)$ assigns to the edge), when assuming that $\mathbb{P}(\rho_i) + \mathbb{P}(\rho_j) = c$ for some constant $2 - 2\psi(\alpha) \leq c \leq 2$, is achieved when $\mathbb{P}(\rho_i)\mathbb{P}(\rho_j)$ is minimal, i.e., (w.l.o.g) when $\mathbb{P}(\rho_i) = 1$ and $\mathbb{P}(\rho_j) = c - 1$. Substituting this assumption in the inequality results in

$$\begin{aligned}\overline{W_{ij}} \leq \quad & 2t_2 + t_2(3/2 - c/2)(\mu_2 - \mu_1) - t_2(c-1)\mu_1 \\ & +2 \sum_{t_2 < t \leq t_3} t\mathbb{P}(\varepsilon_{ijt})(1 + (\mu_2 - \mu_1)/2).\end{aligned}$$

This term is maximal when c is minimal. We therefore substitute the minimal value for c, $c = 2 - 2\psi(\alpha)$, and get

$$\begin{aligned}\overline{W_{ij}} \leq \quad & 2t_2 + t_2(1 - (1 - 2\psi(\alpha))/2)(\mu_2 - \mu_1) - t_2(1 - 2\psi(\alpha))\mu_1 \\ & +2 \sum_{t_2 < t \leq t_3} t\mathbb{P}(\varepsilon_{ijt})(1 + (\mu_2 - \mu_1)/2).\end{aligned}$$

Substituting $\mu_2 = \frac{\mu}{1-2\mu}$ and thus $\mu_2 - \mu_1 = \frac{2\mu^2}{1-2\mu} = 2\mu\mu_2$ we get

$$\overline{W_{ij}} \ \leq \ 2t_2 + t_2\mu(2\mu_2 - (1 - 2\psi(\alpha))(1 + \mu_2)) + 2 \sum_{t_2 < t \leq t_3} t\mathbb{P}(\varepsilon_{ijt})(1 + \mu\mu_2).$$

Using Lemma 4.7 we note that

$$\sum_{t_2 < t \ \leq \ t_3} t\mathbb{P}(\varepsilon_{ijt}) \ \leq \ \sum_{t_2 < t \leq t_3} t(y_{ijt} - y_{ij(t+1)}) \ = \ t_2 y_{ijt_2} + Y_{ij}[t_2, t_3] \ \leq \ \psi(\alpha)t_2 + Y_{ij}[t_2, t_3].$$

Substituting this in the previous inequality concludes the proof. ▮

Bounding $\overline{W_{ij}}$ for E_b^{large} edges: For the edges in E_b^{large} we show the following.

Lemma 4.9 *If $(i,j) \in E_b^{large}$, then*

$$\overline{W_{ij}} \ \leq \ 2t_2 + t_2\mu\left(2\mu_2 - (\mu_2 + 1)(1 - 2\psi(\alpha))\left(1 - \frac{1 - 2\psi(\alpha)}{8}\right)\right).$$

Proof We bound the expected value of $\overline{W_{ij}}$ by relying on the fact that for E_b^{large} edges it holds that

$$\overline{W_{ij}} \ = \ \mathbb{P}(\varphi_1) \cdot E_1 \cdot \bar{E} \ + \ \mathbb{P}(\varphi_2) \cdot E_2 \cdot \bar{E} \ + \ \mathbb{P}(\varphi_3) \cdot E_3 \cdot \bar{E}.$$

We neglect the case of φ_4 and regard it as a sub-case of φ_3, i.e., we refer to φ_3 as the event that by timestep t_2 the edge was covered by *at least* 3 instances. We note that, as adding another covering instance prior to timestep t_2 decreases the expected cover time of the edge, the bound stated above is valid. Note that E_b^{large} edges are surly covered by timestep t_2, so the terms evaluating the expectation given that (i,j) is covered after timestep t_2 are all 0. We assume (w.l.o.g) that $\max(X_{it_2}, X_{jt_2}) = X_{it_2}$. In the following calculations, i' denotes the second instance of vertex i (i.e., $\rho_{i'}$ denotes the event that the second instance of vertex i is chosen by timestep t_2). Note that according to our assumption $\mathbb{P}(\rho_i) = 1$. It will therefore not be explicitly written, unless necessary.

We observe that in this case

$$\begin{aligned}
\mathbb{P}(\varphi_{1,t}) &= \mathbb{P}(\overline{\rho_{i't}} \cap \overline{\rho_{jt}}) &&= (1 - \mathbb{P}(\rho_{i't}))(1 - \mathbb{P}(\rho_{jt})),\\
\mathbb{P}(\varphi_{2,t}) &= \mathbb{P}((\rho_{i't} \cap \overline{\rho_{jt}}) \cup (\rho_{jt} \cap \overline{\rho_{i't}})) &&= \mathbb{P}(\rho_{i't})(1 - \mathbb{P}(\rho_{jt})) + \mathbb{P}(\rho_{jt})(1 - \mathbb{P}(\rho_{i't})),\\
\mathbb{P}(\varphi_{3,t}) &= \mathbb{P}(\rho_{i't} \cap \rho_{jt}) &&= \mathbb{P}(\rho_{i't})\mathbb{P}(\rho_{jt}).
\end{aligned}$$

As in the proof of Lemma 4.8, we combine the above terms to bound the value of $\overline{W_{ij}}$ by

$$\begin{aligned}
\overline{W_{ij}} \leq\ & 2t_2(1 + (\mu_2 - \mu_1)/2)(1 - \mathbb{P}(\rho_{i'}))(1 - \mathbb{P}(\rho_j))\\
& +2t_2(1 + (\mu_2 - \mu_1)/4 - \mu_1/2)(\mathbb{P}(\rho_{i'})(1 - \mathbb{P}(\rho_j)) + \mathbb{P}(\rho_j)(1 - \mathbb{P}(\rho_{i'}))\\
& +2t_2\left(1 + \frac{1}{8}(\mu_2 - \mu_1) - \frac{3}{4}\mu_1\right)\mathbb{P}(\rho_{i'})\mathbb{P}(\rho_j)\\
=\ & 2t_2\\
& +t_2(\mu_2 - \mu_1)(1 - \mathbb{P}(\rho_{i'})/2)(1 - \mathbb{P}(\rho_j)/2)\\
& -t_2\mu_1(\mathbb{P}(\rho_{i'}) + \mathbb{P}(\rho_j) - \mathbb{P}(\rho_{i'})\mathbb{P}(\rho_j)/2).
\end{aligned}$$

Assuming that $\mathbb{P}(\rho_{i'}) + \mathbb{P}(\rho_j) = c$ for some constant c, the term above is maximal when $\mathbb{P}(\rho_{i'})\mathbb{P}(\rho_j)$ is maximal, i.e., when $\mathbb{P}(\rho_{i'}) = \mathbb{P}(\rho_j) = c/2$. The above equation thus becomes

$$\overline{W_{ij}} \ \leq\ 2t_2 + t_2(\mu_2 - \mu_1)(1 - c/4)^2 - t_2\mu_1(c - c^2/8).$$

Substituting $\mu_2 - \mu_1 = 2\mu\mu_2$ (and further simplifying) yields

$$\begin{aligned}
\overline{W_{ij}} &\leq 2t_2 + t_2\mu(2\mu_2(1 - c/4)^2 - c(1 - c/8))\\
&= 2t_2 + t_2\mu(2\mu_2 - (\mu_2 + 1)c(1 - c/8)).
\end{aligned}$$

Note that since $0 \leq y_{ijt_2} \leq \psi(\alpha)$, it follows that $1 - 2\psi(\alpha) \leq \mathbb{P}(\rho_{i'}) + \mathbb{P}(\rho_j) \leq 1$, implying that $1 - 2\psi(\alpha) \leq c \leq 1$. The bound is maximal when c is minimal, yielding

$$\overline{W_{ij}} \ \leq\ 2t_2 + t_2\mu\left(2\mu_2 - (\mu_2 + 1)(1 - 2\psi(\alpha))\left(1 - \frac{1 - 2\psi(\alpha)}{8}\right)\right). \qquad \blacksquare$$

A combined bound on E_b edges: We note that the bound of Lemma 4.9 can be rewritten as

$$\overline{W_{ij}} \ \leq\ 2t_2 + t_2\mu(2\mu_2 - (\mu_2 + 1)(1 - 2\psi(\alpha))) + t_2\mu(1 + \mu_2)\frac{(1 - 2\psi(\alpha))^2}{8},$$

and therefore, Lemmas 4.8 and 4.9 can be combined into the following.

Corollary 4.10 *If* $(i,j) \in E_b$ *then*

$$\begin{aligned}\overline{W_{ij}} &\leq 2t_2 + t_2\mu(2\mu_2 - (\mu_2+1)(1-2\psi(\alpha))) \\ &\quad + \max\left(t_2\mu(1+\mu_2)\frac{(1-2\psi(\alpha))^2}{8}, 2(1+\mu\mu_2)(t_2\psi(\alpha)+Y_{ij}[t_2,t_3])\right)\end{aligned}$$

Lower bounding W_{ij} on E_b edges: We bound from below the value that Π can assign to an edge $(i,j) \in E_b$. This bound is used to obtain an upper bound on $\overline{R_{ij}}$ on such edges. We need this bound to get a term that is dependent on t_1, t_2 and t_3. We start by showing the following.

Lemma 4.11 *For all* $(i,j) \in G$,

$$W_{ij} \geq (1-2\psi(\alpha))t_1 + \psi(\alpha)t_2 + Y_{ij}[t_2,t_3].$$

Proof Express W_{ij} as

$$W_{ij} = Y_{ij}[0,t_3] = Y_{ij}[0,t_1] + Y_{ij}[t_1,t_2] + Y_{ij}[t_2,t_3].$$

Note that by the definition of t_1, $y_{ijt} \geq 1-\psi(\alpha)$ for all $t < t_1$. Also note that by the definition of t_2, $y_{ijt} \geq \psi(\alpha)$ for all $t_1 \leq t < t_2$. Thus

$$\begin{aligned}W_{ij} &\geq \sum_{0\leq t<t_1}(1-\psi(\alpha)) + \sum_{t_1\leq t<t_2}\psi(\alpha) + Y_{ij}[t_2,t_3] \\ &= (1-\psi(\alpha))t_1 + \psi(\alpha)(t_2-t_1) + Y_{ij}[t_2,t_3] \\ &= (1-2\psi(\alpha)t_1 + \psi(\alpha)t_2 + Y_{ij}[t_2,t_3]. \quad \blacksquare\end{aligned}$$

Edges $(i,j) \in E_b$ satisfy $t_1 > (1-\beta)t_2$. Therefore we get the following.

Corollary 4.12 *If* $(i,j) \in E_b$ *then*

$$W_{ij} \geq (1-2\psi(\alpha))(1-\beta)t_2 + \psi(\alpha)t_2 + Y_{ij}[t_2,t_3]$$

Bounding $\overline{R_{ij}}$ on E_b edges: To bound $\overline{R_{ij}}$ on edges in E_b we consider the two cases analyzed in Lemma 4.8 and in Lemma 4.9. When considering the case of Lemma 4.8 we get

Lemma 4.13 *If* $(i,j) \in E_b^{small}$ *then*

$$\overline{R_{ij}} \leq \frac{2(1-\mu)(-2\gamma\beta^2 + 2(\gamma-1)\beta + \mu(2\gamma\beta^2 - (\gamma-2)\beta - 3) - \gamma + 3)}{(2\mu-1)((2\beta-1)\gamma-1)}.$$

Proof Dividing the bound from Lemma 4.8 by the bound from Corollary 4.12 yields

$$\overline{R_{ij}} \;\leq\; \frac{2t_2 + t_2\mu(2\mu_2 - (1-2\psi(\alpha))(1+\mu_2)) + 2(1+\mu\mu_2)t_2\psi(\alpha) + 2(1+\mu\mu_2)Y_{ij}[t_2,t_3]}{(1-2\psi(\alpha))(1-\beta)t_2 + \psi(\alpha)t_2 + Y_{ij}[t_2,t_3]}.$$

We note that (assuming $\overline{R_{ij}} < 2$) this term is maximal when W_{ij} is minimal, but $\frac{Y_{ij}[t_2,t_3]}{W_{ij}}$ is maximal. As $(i,j) \in E_b$ implies that $\frac{Y_{ij}[t_2,t_3]}{W_{ij}} \leq \beta$, and reusing the bound form Corollary 4.12, we get that

$$\frac{Y_{ij}[t_2,t_3]}{(1-2\psi(\alpha))(1-\beta)t_2 + \psi(\alpha)t_2 + Y_{ij}[t_2,t_3]} \;=\; \beta,$$

that results in

$$Y_{ij}[t_2,t_3] \;=\; \frac{\beta}{1-\beta}((1-2\psi(\alpha))(1-\beta)t_2 + \psi(\alpha)t_2).$$

Substituting $Y_{ij}[t_2,t_3]$ in the definition of $\overline{R_{ij}}$ (and canceling t_2) yields

$$\overline{R_{ij}} \;\leq\; \frac{(1-\beta)(2 + \mu(2\mu_2 - (1-2\psi(\alpha))(1+\mu_2)) + 2(1+\mu\mu_2)\psi(\alpha))}{(1-2\psi(\alpha))(1-\beta) + \psi(\alpha)} + 2\beta(1+\mu\mu_2).$$

Substituting μ_2, $\psi(\alpha)$ and simplifying yields

$$\overline{R_{ij}} \;\leq\; \frac{2(1-\mu)(-2\gamma\beta^2 + 2(\gamma-1)\beta + \mu(2\gamma\beta^2 - (\gamma-2)\beta - 3) - \gamma + 3)}{(2\mu-1)((2\beta-1)\gamma-1)} \quad \blacksquare$$

When considering the case analyzed in Lemma 4.9 we get

Lemma 4.14 *If $(i,j) \in E_b^{large}$ then*

$$\overline{R_{ij}} \;\leq\; \frac{(\mu-1)(\mu(4\alpha+8\gamma+15)-16)}{4(2\mu-1)((2\beta-1)\gamma-1)}.$$

Proof Dividing the bound from Lemma 4.9 by the bound from Corollary 4.12 yields

$$\overline{R_{ij}} \;\leq\; \frac{2t_2 + t_2\mu\left(2\mu_2 - (\mu_2+1)(1-2\psi(\alpha))\left(1 - \frac{1-2\psi(\alpha)}{8}\right)\right)}{(1-2\psi(\alpha))(1-\beta)t_2 + \psi(\alpha)t_2 + Y_{ij}[t_2,t_3]}.$$

This value is maximal when $Y_{ij}[t_2,t_3]$ is minimal, i.e., when $Y_{ij}[t_2,t_3] = 0$, which yields (when simplifying)

$$\overline{R_{ij}} \;\leq\; \frac{2 + \mu\left(2\mu_2 - (\mu_2+1)(1-2\psi(\alpha))\left(1 - \frac{1-2\psi(\alpha)}{8}\right)\right)}{(1-2\psi(\alpha))(1-\beta) + \psi(\alpha)}.$$

Substituting μ_2 and $\psi(\alpha)$ yields

$$\overline{R_{ij}} \;\leq\; \frac{(\mu-1)(\mu(4\alpha+8\gamma+15)-16)}{4(2\mu-1)((2\beta-1)\gamma-1)}. \quad \blacksquare$$

Combining Lemmas 4.13, 4.14 and 2.6 yields:

Corollary 4.15 *If* $(i,j) \in E_b$ *then*

$$\overline{R_{ij}} \leq \max \Bigg(\frac{2(1-\mu)(-2\gamma\beta^2 + 2(\gamma-1)\beta + \mu(2\gamma\beta^2 - (\gamma-2)\beta - 3) - \gamma + 3)}{(2\mu-1)((2\beta-1)\gamma-1)}, \frac{(\mu-1)(\mu(4\alpha+8\gamma+15)-16)}{4(2\mu-1)((2\beta-1)\gamma-1)} \Bigg)$$

and $\widehat{R_{ij}} \leq 2$.

We note that for certain choices of the values of the parameters α, β and μ, the bound on $\overline{R_{ij}}$ implied by the corollary is strictly smaller than 2. In particular, when assigning $\alpha = 0.01$, $\beta = 0.01$, and $\mu = 0.2$ we get that the second case is the maximum and is equal to 1.943.

4.6 Bounding the approximation ratio of $\min(\textsf{RR}, \textsf{MRR}(\mu))$

We need to get a term for the global approximation bound depending on α, β and μ. We proceed to find an assignment maximizing this term. We use the definitions of E_a and E_b to devise a way to get an approximation ratio strictly better than 2 for MSVC.

Lemma 4.16 *For any input graph* G*, and solutions* $\Pi(G)$*,* $\mathit{RR}(G)$ *and* $\mathit{MRR}(\mu, G)$*, it holds that*

$$\min\left(\frac{\mathit{RR}(G)}{\Pi(G)}, \frac{\mathit{MRR}(\mu, G)}{\Pi(G)}\right) \;\leq\; 2 - \frac{\varepsilon_1\varepsilon_2}{\varepsilon_1 + \varepsilon_2 + \frac{2\mu^2}{1-2\mu}} \;<\; 2$$

Proof Use the output of Π to obtain E_a and E_b.

Define $W_H = \sum_{(i,j)\in H} W_{ij}$, where $H = E_a$ or E_b. In a similar manner define $\widehat{W_H} = \sum_{(i,j)\in H} \widehat{W_{ij}}$ and $\overline{W_H} = \sum_{(i,j)\in H} \overline{W_{ij}}$.

Let $0 \leq Z \leq 1$ s.t., $W_{E_a} = Z \sum_{(i,j)\in G} W_{ij} = Z\Pi(G)$. Note that $W_{E_b} = (1-Z)\Pi(G)$.

According to Lemma 4.6 and Corollary 4.15 we get that

$$\begin{aligned} \mathsf{RR}(G) &= \widehat{W_{E_a}} + \widehat{W_{E_b}} \\ &\leq (2-\varepsilon_1)W_{E_a} + 2W_{E_b} \\ &= ((2-\varepsilon_1)Z + 2(1-Z))\Pi(G), \end{aligned}$$

and that

$$\begin{aligned}\mathsf{MRR}(\mu, G) &= \overline{W_{E_a}} + \overline{W_{E_b}} \\ &\leq \left(2 + \frac{2\mu^2}{1-2\mu}\right) W_{E_a} + (2-\varepsilon_2) W_{E_b} \\ &= \left(\left(2 + \frac{2\mu^2}{1-2\mu}\right) Z + (2-\varepsilon_2)(1-Z)\right) \Pi(G).\end{aligned}$$

Finding the value of Z for which the two terms are equal $Z_{\max}$ yields an upper bound on the approximation ratio, since if $Z > Z_{\max}$ $\mathsf{RR}(G)/\Pi(G)$ is reduced, and if $Z < Z_{\max}$ $\mathsf{MRR}(\mu, G)/\Pi(G)$ is reduced. We therefore get that $Z_{\max}$ satisfies

$$((2-\varepsilon_1)Z_{\max} + 2(1-Z_{\max}))\Pi(G) = \left(\left(2 + \frac{2\mu^2}{1-2\mu}\right) Z_{\max} + (2-\varepsilon_2)(1-Z_{\max})\right) \Pi(G),$$

which yields

$$Z_{\max} = \frac{\varepsilon_2}{\varepsilon_1 + \varepsilon_2 + \frac{2\mu^2}{1-2\mu}}.$$

The approximation ratio (B) is thus bounded by

$$B \leq (2-\varepsilon_1)Z_{\max} + 2(1-Z_{\max}) = 2 - \varepsilon_1 Z_{\max} = 2 - \frac{\varepsilon_1\varepsilon_2}{\varepsilon_1 + \varepsilon_2 + \frac{2\mu^2}{1-2\mu}} \qquad \blacksquare$$

Finding an optimal assignment for α, β and μ (that minimizes the approximation ratio) is done numerically. It results in an approximation ratio of 1.9999460023987983, with

$$\begin{aligned}\mu &\to 0.1599419805972847, \\ \alpha &\to 0.010241449815300075, \\ \beta &\to 0.020142468138612125\end{aligned}$$

5 Further improvements

There are a few ideas we believe can be investigated to improve the approximation ratio. In particular, we note that the probability of moving vertices to previous or latter timesteps is arbitrarily chosen as 1/2. Adding it as a parameter to the analysis and solving for its optimal value may improve the results. Also, one might consider defining the reorganization step using a more sophisticated distribution than the binary one we used. Finally, there are a few "slacks" in our analysis of MRR that can be reduced by a more careful analysis. For example, adding notions for timesteps at which y_{ijt} crosses values in between t_1 and t_2 can be used to either increase the lower bound on W_{ij} or decrease the upper bound on $\overline{W_{ij}}$, thus reducing $\overline{R_{ij}}$.

The other main technique that can be applied and analyzed is semidefinite programming. A motivating example of the usage of semidefinite programming is presented in [4], were results from [3] are applied to construct an algorithm achieving a better approximation ratio for min sum vertex cover on regular graphs. The subject of applying semidefinite programming techniques for approximation algorithms is presented in [5].

References

[1] A. Bar-Noy, M. Bellare, M. Halldorsson, H. Shachnai, T. Tamir. On chromatic sums and distributed resource allocation. *Information and Computation*, 140:183–202, 1998.

[2] S. Burer and R. Monteiro. A projected gradient algorithm for solving the maxcut SDP relaxation. *Optimization Methods and Software* 15, (2001), 175–200.

[3] U. Feige and M. Langberg. Approximation algorithms for maximization problems arising in graph partitioning. *J. of Algorithms* 41, (2001), 174–211.

[4] U. Feige, L. Lovasz and P. Tetali. Approximating Min-sum Set Cover. *Algorithmica* 40, (2004), 219–234

[5] M.X. Goemans and D.P. Williamson. Improved Approximation Algorithms for Maximum Cut and Satisfiability Problems Using Semidefinite Programming. *J. ACM* 42, (1995), 1115–1145.

[6] M. X. Goemans and D. P. Williamson. New 3/4-Approximation Algorithm for MAX SAT. *SIAM J. on Discrete Mathematics* 7, (1994), 656–666.