Approximability and Inapproximability for Maximum k-Edge-Colored Clustering Problem*

Yousef M. Alhamdan[1][0000−0002−3064−5884] and Alexander Kononov[2][0000−0001−6144−0251]

[1] Mathematics and Mechanics Department, Novosibirsk State University
yousify32@gmail.com

[2] Mathematics and Mechanics Department, Novosibirsk State University and Sobolev Institute of Mathematics
alvenko@math.nsc.ru

Abstract. We consider the Max k-Edge-Colored Clustering problem (abbreviated as MAX-k-EC). We are given an edge-colored graph with k colors. Each edge of the graph has a positive weight. We seek to color the vertices of the graph so as to maximize the total weight of the edges which have the same color at their extremities. The problem was introduced by Angel et al. [2]. We give a polynomial-time algorithm for MAX-k-EC with an approximation factor $\frac{49}{144} \approx 0.34$, which significantly improves the best previously known factor $\frac{7}{23} \approx 0.304$, obtained by Ageev and Kononov [1]. We also present an upper bound of $\frac{241}{248} \approx 0.972$ on the in-approximability of MAX-k-EC. This is the first inapproximability result for this problem.

Keywords: Approximation algorithm analysis · Problems, reductions and completeness · Colored clustering

1 Introduction

In this paper, we study the Max k-Edge-Colored Clustering problem introduced by Angel et al. [2]. Given an undirected graph $G = (V, E)$ with colors of edges $c : E \to \{1, \dots, k\}$ and weights $w : E \to Q^+$. Our goal is to color vertices of G so as to maximize the total weight of edges which have the same color at their extremities. Denote the set of colors by $\mathcal{C}$, i.e., $\mathcal{C} = \{1, \dots, k\}$. Given a vertex coloring, we say that an edge of G is *stable* if both its extremities have the same color as the color of the edge[3]. Thus, we are interested to assign a color from $\mathcal{C}$ to each vertex of G so as to maximize the total weight of stable edges.

As observed by Cai and Leung [5], the MAX-k-EC problem can be considered as the optimization counterpart of the Vertex-Monochromatic Subgraph problem or the Alternating Path Removal problem. In the Vertex-Monochromatic

* This work was appeared as part of the first author's MSc thesis.

[3] Note that, "stable" is defined as "matched" in Angel et al. [2] and Ageev and Kononov [1]. But, we follow Cai and Leung [5] definition since it makes sense.

Subgraph problem it is required to find the largest subgraph where every vertex has one color for its incident edges or, what is the same, remove the smallest number of edges to destroy all the alternating paths, as required in the Alternating Path Removal problem. Although a huge number of papers are devoted to monochromatic subgraphs and alternating paths [4, 9], Angel et al. [2] were the first to consider the optimization problem concerning these concepts.

The MAX-k-EC problem is also a natural generalization of Maximum Weight Matching Problem. Indeed, if each edge has its own color the problem coincides with the edge packing problem which is equivalent to finding a maximum weight matching. Our problem can also be considered as an extension of the centralized version of the information-sharing model introduced by Kleinberg and Ligett [10]. In their model, the edges are not colored. Two adjacent vertices share information only if they are colored with the same color but some pairs of vertices are forbidden to color in one color. As Kleinberg and Ligett mention, it is interesting to expand their model by considering different categories of information. In the MAX-k-EC problem every edge-color corresponds to a different information category and two adjacent vertices share information if their color is the same as the color of the edge that connects them.

On the other hand, MAX-k-EC is a special case of the combinatorial allocation problem [6]. We associate each color with a player and each vertex with an item, where items have to be allocated to competing players by a central authority, with the goal of maximizing the total utility provided to the players. Every player has utility functions derived from the different subsets of items. In the most general setting, utility functions are arbitrary functions satisfying the monotonicity property. Feige and Vondrak [6] consider subadditive, fractional subadditive and submodular functions and present constant factor approximation algorithms. It is easy to check that in our problem the function is supermodular. For the general supermodular function no polynomial time algorithm can find an allocation within a factor $n^{(1-\varepsilon)}$ from optimum for any $\varepsilon > 0$ unless NP=ZPP.

Angel et al. [2, 3] presented a polynomial-time algorithm for the MAX-k-EC problem on edge-bicoloured graphs by a reduction to the maximum independent set problem on bipartite graphs. Moreover, they showed the strongly NP-hardness of the problem for edge-tricoloured bipartite graphs. Recently, Cai and Leung [5] expanded the last result and proved that MAX-k-EC is NP-hard in the strong sense even on edge-tricoloured planar bipartite graphs of maximum degree four. Angel et al. [2, 3] obtained an LP-based approximation algorithm for the MAX-k-EC problem and showed that it finds a solution such that the weight of stable edges is at least $\frac{1}{e^2}$ from the optimum. Later, Ageev and Kononov [1] gave a better analysis of the same algorithm of [2, 3] and improved the ratio from $\frac{1}{e^2} = 0.135$ to 0.25. They also presented a $\frac{7}{23}$-approximation algorithm based on the same LP introduced in [2, 3]. Cai and Leung [5] derived two FPT algorithms for the MAX-k-EC problem when they consider the numbers of stable edges and unstable edges, respectively, as a fixed parameter.

1.1 Our Contributions

In Section 2 we present a new algorithm for the MAX-k-EC problem with approximation ratio $\frac{49}{144} \approx 0.34$. As in [2, 3], we formulate the problem as an integer linear program and develop a randomized rounding scheme for the linear programming relaxation. We design a two-phase scheme, where the first phase selects a set of desired edges randomly and independently for each color, while the second phase colors vertices, taking into account the selection of the edges made in the first phase.

In Section 3 we provide an inapproximability result showing that there is no ρ-approximation algorithm for the MAX-k-EC problem for constant $\rho > \frac{241}{248}$ unless P=NP. Our proof uses an L-reduction from the well-known MAX-E3-SAT problem.

2 Two-phase Randomized Approximation Algorithm

Consider two sets of variables z_e, $e \in E$ and x_{vi}, $v \in V$, $i \in \mathcal{C}$, where $z_e = 1$ if both endpoints of e are colored with the same color as e and $z_e = 0$ otherwise and $x_{vi} = 1$ if v is colored with color i and $x_{vi} = 0$ otherwise. Angel et al. [2, 3] introduced the following integer linear program (ILP) for MAX-k-EC.

$$\text{maximize} \sum_{e \in E} w_e z_e \tag{1}$$

$$\text{subject to} \sum_{i \in \mathcal{C}} x_{vi} = 1, \qquad \forall v \in V \tag{2}$$

$$z_e \leq \min\{x_{vc(e)}, x_{uc(e)}\} \qquad \forall e = [v, u] \in E \tag{3}$$

$$x_{vi}, z_e \in \{0, 1\}, \qquad \forall v \in V, i \in \mathcal{C}, e \in E \tag{4}$$

The first set of constraints ensures that each vertex is colored in exactly one color, and the second ensures that an edge e is stable if its color is the same as the color of both its extremities.

The LP-relaxation (LP) of (1)-(4) is obtained by replacing the constraints $x_{vi} \in \{0, 1\}$ and $z_e \in \{0, 1\}$ by $x_{vi} \geq 0$ and $z_e \geq 0$, respectively. The following two-phase algorithm was considered in [2, 1]. In the first phase, it starts with solving LP and then works in k iterations, by considering each color i, $1 \leq i \leq k$, independently from the others. For each color, the algorithm picks r at random in $(0, 1)$ and chooses all edges of this color with $z_e^* \geq r$. When an edge is chosen, this means that both its endpoints get the color of this edge. Since in general a vertex can be adjacent to differently colored edges, it may get more than one colors. In the second phase, the algorithm chooses randomly one of these colors. Denote by $\lambda_v(l, c)$ the probability with which the algorithm chooses the color c if l colors were assigned to v at the first phase of the algorithm. We present the algorithm below.

Algorithm 1 Algorithm 2-PHASE

```
Phase I:
Solve LP and let z_e^* be the values of variables z_e.
for each color c ∈ C do
    Let r be a random value in [0,1].
    Choose the c-colored edges e with z_e^* ≥ r and give color c to both of e's endpoints.
end for
Phase II:
for each vertex v ∈ V do
    Let vertex v got l colors.
    assign randomly one of l colors to v, each with the probability λ_v(l, c).
end for
```

Remind that in the second phase the algorithm from [1] picks each color with equal probability, i.e. $\lambda_v(l,c) = \frac{1}{l}$ for all colors assigned to v at the first phase of the algorithm. In our algorithm we change this property. Let (x^*, z^*) be an optimal solution of the LP. Following [1] we say that an edge e is *big* if $z_e^* > \frac{1}{2}$; otherwise an edge e is *small*. We say that a vertex v is *heavy* if it is extrimity of at least one big edge; otherwise we say that vertex v is *light*. We say that a color i is a *heavy* color for a vertex v if v is incident to an i-colored big edge, otherwise a color i is a *light* color for a vertex v. We note that each vertex has at most one heavy color. If the vertex v got two colors: a heavy color i and a light color j then we set $\lambda_v(2,i) = \frac{1}{3}$ and $\lambda_v(2,j) = \frac{2}{3}$ else we set $\lambda_v(l,c) = \frac{1}{l}$ for all l colors assigned to v at the first phase of the algorithm. We call this algorithm *2-Phase with Heavy and Light Vertices* (2-PHLV). This simple idea leads to an improved approximation guarantee for MAX-k-EC, even though it contradicts the intuition; since in ***Phase II*** we increase the probability of picking light colors. Although at first glance we made minor changes to the algorithm from [1], we will have to refine the analysis of new algorithm because the previous one strongly relied on the specifics of the old algorithm.

2.1 Analysis

In this subsection, we give a worst-case analysis of algorithm 2-PHLV. Let X_{vi} denote the event where vertex v gets color i after ***Phase I*** of the algorithm. Since the first phase of the algorithm 2-PHLV coincides with the first phases of the algorithms RR and RR2, presented in [2] and [1], respectively, the following simple statements are valid.

Lemma 1. *[2] For any edge $e \in E$, the probability that e is chosen in* ***Phase I*** *is z_e^*.*

Lemma 2. *[2] For every vertex $v \in V$ and for all $i \in \mathcal{C}$ we have:*

$$Pr[X_{vi}] = \max\{z_e^* : e = [v,u] \in E \;\&\; c(e) = i\}.$$

Lemma 3. *[2] For every vertex $v \in V$, $\sum_{i \in \mathcal{C}} Pr[X_{vi}] \leq 1$.*

Remind that the vertex v can get several colors after ***Phase I***. However, in general this number will be small. Let Y_{vi} denote the event where vertex v is colored with i after ***Phase II*** of the algorithm. The following lemmas give a lower bound for the probability that color i was assigned to vertex v in ***Phase II***. We consider three possible cases: a heavy vertex and a heavy color, a heavy vertex and a light color, a light vertex and a light color.

Lemma 4. *Assume that a heavy vertex v gets a heavy color q in* ***Phase I*** *of Algorithm 2-PHLV, then $Pr[Y_{vq}|X_{vq}] \geq \frac{2}{3}$.*

Proof. The probability that a vertex v is colored with a color q in ***Phase II*** depends on how many colors a vertex v received in ***Phase I***. Without loss of generality, assume that the edges with colors $1, \dots, t$ and q are incident to the vertex v. By the law of total probability we have

$$Pr[Y_{vq}|X_{vq}] \geq \prod_{i=1}^{t}(1 - Pr[X_{vi}]) + \frac{1}{3}\sum_{i=1}^{t} Pr[X_{vi}] \prod_{j \neq i}(1 - Pr[X_{vj}]) \qquad (5)$$

The first term is the probability that a vertex v is colored with color q and no additional color is chosen. The second term is the probability that the vertex v is colored with color q under the condition that it received one additional color in ***Phase I***. Since the color q is heavy, we have $\lambda_v(2, q) = \frac{1}{3}$ and the vertex v will be colored in color q with probability $1/3$. We drop all the remaining terms of the formula because they are equal to zero in the worst case.

To simplify computations we set $X_i = Pr[X_{vi}]$ and consider the right-hand-size of (5) as a function f_{vq} of variables $X_1, X_2, ..., X_t$. We have

$$f_{vq} = \prod_{i=1}^{t}(1 - X_i) + \frac{1}{3}\sum_{i=1}^{t} X_i \prod_{j \neq i}(1 - X_j).$$

Taking into account that color q is heavy, from (2) we have $\sum_{i=1}^{t} X_i \leq \frac{1}{2}$. In order to obtain a lower bound for $Pr[Y_{vq}|X_{vq}]$, consider the minimum value of f_{vq} over all choises of X_i subject to the constraint $\sum_{i=1}^{t} X_i \leq \frac{1}{2}$. Putting the first two variables out of the summation and the product, we get

$$f_{vq} = (1 - X_1)(1 - X_2)\prod_{i=3}^{t}(1 - X_i) + \frac{1}{3}(X_1(1 - X_2) + X_2(1 - X_1))\prod_{i=3}^{t}(1 - X_i) + \frac{1}{3}(1 - X_1)(1 - X_2)\sum_{i=3}^{t} X_i \prod_{j \geq 3, j \neq i}(1 - X_j) \qquad (6)$$

Then from (6) it follows that

$$f_{vq} = (1 - \frac{2}{3}X_1 - \frac{2}{3}X_2)\prod_{i=3}^{t}(1 - X_i) + \frac{1}{3}(1 - X_1 - X_2)\sum_{i=3}^{t} X_i \prod_{j\geq 3, j\neq i}(1 - X_j)$$
$$+ \frac{1}{3}X_1X_2(\prod_{i=3}^{t}(1 - X_i) + \sum_{i=3}^{t} X_i \prod_{j\geq 3, j\neq i}(1 - X_j)) \quad (7)$$

Consider f_{vq} as a function of two variables X_1 and X_2. Assume that $X_1 + X_2 = \gamma$, where $\gamma \leq \frac{1}{2}$ is a constant. Let $X_1 \geq X_2 > 0$. If we increase X_1 and decrease X_2 by δ, $0 < \delta \leq X_2$, then the first two terms of (7) do not change and the last term decreases and therefore the function f_{vq} decreases as well. It follows that the minimum of f_{vq} is attained at $X_1 = \gamma$ and $X_2 = 0$. By repeating this argument we get that the minimum of f_{vq} is attained when $X_1 = \frac{1}{2}$ and $X_i = 0$, $i = 2, \ldots, t$. Finally, we get $Pr[Y_{vq}|X_{vq}] \geq f_{vq} \geq \frac{2}{3}$.

Lemma 5. *Assume that a heavy vertex v gets a light color q in **Phase I** of Algorithm 2-PHLV, then $Pr[Y_{vq}|X_{vq}] \geq \frac{5}{8}$.*

Proof. The probability that a vertex v is colored with a color q in ***Phase II*** depends on what other colors were also chosen for it in ***Phase I***. Without loss of generality, assume that the edges with colors $1, \ldots, t$ and q are incident to the vertex v and let color 1 be the heavy color. By the law of total probability we have

$$Pr[Y_{vq}|X_{vq}] \geq \prod_{i=1}^{t}(1 - Pr[X_{vi}]) + \frac{2}{3}Pr[X_{v1}]\prod_{i=2}^{t}(1 - Pr[X_{vi}])$$
$$+ \frac{1}{2}(1 - Pr[X_{v1}])\sum_{i=2}^{t} Pr[X_{vi}] \prod_{j\geq 2, j\neq i}(1 - Pr[X_{vj}])$$
$$+ \frac{1}{3}\sum_{i=1}^{t}\sum_{j=i+1}^{t} Pr[X_{vi}]Pr[X_{vj}] \prod_{\acute{i}=1, \acute{i}\neq i, \acute{i}\neq j}(1 - Pr[X_{v\acute{i}}])$$

The first term is the probability that a vertex v is colored with color q and no additional color is chosen. The second term is the probability that the vertex v is colored with color q under the condition that it receives an additional heavy color in ***Phase I***. Since the vertex q is light, we have $\lambda_v(2, q) = \frac{2}{3}$ and the vertex v will be colored in color q with probability 2/3. The third term is the probability that the vertex v is colored with color q under the condition that it receives exactly one additional light color in ***Phase I***. In this case, the vertex v will be colored in color q with probability 1/2. The last term is the probability that the vertex v is colored with color q under the condition that for v two more colors were chosen. We drop all the remaining terms of the formula because they are equal to zero in the worst case.

By setting $A = \prod_{i=3}^{t}(1 - Pr[X_{vi}])$, $B = \sum_{i=3}^{t} Pr[X_{vi}] \prod_{j\geq 3, j\neq i}(1 - Pr[X_{vj}])$ and

$C = \sum_{i=3}^{t} \sum_{j=i+1}^{t} Pr[X_{vi}] Pr[X_{vj}] \prod_{\acute{i} \geq 3, \acute{i} \neq i, \acute{i} \neq j} (1 - Pr[X_{v\acute{i}}])$ we can rewrite this expression as

$$Pr[Y_{vq}|X_{vq}] \geq (1 - Pr[X_{v1}])(1 - Pr[X_{v2}])A + \frac{2}{3} Pr[X_{v1}](1 - Pr[X_{v2}])A +$$
$$\frac{1}{2}(1 - Pr[X_{v1}]) Pr[X_{v2}]A + \frac{1}{2}(1 - Pr[X_{v1}])(1 - Pr[X_{v2}])B + \frac{1}{3} Pr[X_{v1}] Pr[X_{v2}]A$$
$$+ \frac{1}{3}(Pr[X_{v1}](1 - Pr[X_{v2}]) + (1 - Pr[X_{v1}]) Pr[X_{v2}])B + \frac{1}{3}(1 - Pr[X_{v1}])(1 - Pr[X_{v2}])C.$$

Discarding the last term and setting $X_i = Pr[X_{vi}]$ we get

$$Pr[Y_{vq}|X_{vq}] \geq f_{vq} \doteq (1 - X_1)(1 - X_2)A + \frac{2}{3} X_1(1 - X_2)A + \frac{1}{2}(1 - X_1)X_2 A$$
$$+ \frac{1}{2}(1 - X_1)(1 - X_2)B + \frac{1}{3} X_1 X_2 A + \frac{1}{3}(X_1(1 - X_2) + (1 - X_1)X_2)B.$$

In order to obtain a lower bound for $Pr[Y_{vq}|X_{vq}]$, we first show that the minimum of f_{vq} is attained when $X_1 = \frac{1}{2}$. After multiplying the terms with each other, we get

$$f_{vq} = (1 - \frac{1}{3}X_1 - \frac{1}{2}X_2 + \frac{1}{6}X_1 X_2)A + (\frac{1}{2} - \frac{1}{6}X_1 - \frac{1}{6}X_2 - \frac{1}{6}X_1 X_2)B$$
$$= (1 - \frac{1}{3}X_1 - \frac{1}{3}X_2)A + (\frac{1}{2} - \frac{1}{6}X_1 - \frac{1}{6}X_2)B - \frac{1}{6}(X_1 X_2 B + X_2(1 - X_1)A).$$

Let us consider f_{vq} as a function of two variables X_1 and X_2. Assume that $X_1 + X_2 = \gamma$. Since color 1 is heavy then $\frac{1}{2} \leq X_1 \leq \gamma$ and $X_2 \leq \frac{1}{2}$. If we decrease X_1 and increase X_2 by δ, $0 < \delta \leq X_1 - \frac{1}{2}$, then the expression $X_1 X_2 B + X_2(1 - X_1)A$ increases. It follows that f_{vq} reaches a minimum when $X_1 = \frac{1}{2}$.

Now, substitute X_1 by $\frac{1}{2}$. Thus, we obtain

$$f_{vq} \geq (\frac{5}{6} - \frac{5}{12}X_2) \prod_{i=3}^{t} (1 - X_i) + (\frac{5}{12} - \frac{1}{4}X_2) \sum_{i=3}^{t} X_i \prod_{j \geq 3, j \neq i} (1 - X_j). \quad (8)$$

Rewrite the right-hand side of (8) as

$$f_{vq} = (\frac{5}{6} - \frac{5}{12}X_2)(1 - X_3) \prod_{i=4}^{t} (1 - X_i) + (\frac{5}{12} - \frac{1}{4}X_2) X_3 \prod_{i=4}^{t} (1 - X_i)$$
$$+ (\frac{5}{12} - \frac{1}{4}X_2)(1 - X_3) \sum_{i=4}^{t} X_i \prod_{j=4, j \neq i}^{t} (1 - X_j) \quad (9)$$

Discarding the last term in (9) we get

$$f_{vq} \geq \frac{5}{6}(1 - \frac{1}{2}X_2 - \frac{1}{2}X_3 + \frac{1}{5}X_2 X_3) \prod_{i=4}^{t} (1 - X_i).$$

Taking into account that $X_2 \leq \frac{1}{2}$ and $X_3 \leq \frac{1}{2}$, we finally obtain $f_{vq} \geq \frac{5}{8}$.

The following result directly follows from Lemma 4(c) in [1].

Lemma 6. *Assume that a light vertex v gets a color q in **Phase I** of Algorithm 2-PHLV, then $Pr[Y_{vq}|X_{vq}] \geq \frac{7}{12}$.*

Suppose that $e = (u, v)$ has a color c and it is chosen in the first phase of Algorithm 2-PHLV and let $Pr[e \text{ is stable}]$ denote the probability that both vertices v and u get the color c. We want to prove that

$$Pr[e \text{ is stable}] \geq Pr[Y_{uc}]Pr[Y_{vc}]. \tag{10}$$

Similar results were proven for the algorithms RR and RR2, presented in [2] and [1], respectively. Algorithm 2-PHLV differs from algorithms RR and RR2 only by the choice of probabilities $\lambda_v(l, c)$ in the second phase of the algorithm. Unfortunately, the previous proofs of (10) are not suitable for our algorithm. Here, we introduce a sufficient condition for probabilities $\lambda_v(l, c)$ such that (10) holds. The Algorithm 2-PHASE specifies values of $\lambda_v(l, c)$ only if the edge of color c incident to the vertex v was chosen in the first phase of the algorithm. Without loss of generality, we assume that $\lambda_v(l, c) = 0$ otherwise.

Lemma 7. *Let $e = (u, v)$ has a color c and it is chosen in the first phase of Algorithm 2-PHASE. If the sequences $\lambda_v(1, c), \ldots, \lambda_v(k, c)$ and $\lambda_u(1, c), \ldots, \lambda_u(k, c)$ are non-increasing then $Pr[e \text{ is stable}] \geq Pr[Y_{uc}|X_{uc}]Pr[Y_{vc}|X_{vc}]$.*

Due to space limitation, the proof of the lemma is removed in this version.

Theorem 1. *The expected approximation ratio of Algorithm 1 is bounded by $\frac{49}{144}$.*

Proof. Let OPT denote the sum of the weights of the stable edges in an optimal solution. Since z^* is an optimal solution of the LP, we have $OPT \leq \sum_{e \in E} w_e z_e^*$.

Consider an edge $e \in E$ is chosen in ***Phase I*** of Algorithm 2-PHLV. This occurs with probability z_e^* by lemma 1. Suppose an edge $e = [u, v]$ has a color c, then by Lemma 7, the probability that the both endpoints of e are colored with c at least $Pr[Y_{vc}|X_{vc}]Pr[Y_{uc}|X_{uc}]$. Lemmata 4-6 imply that the expected contribution of the edge e is at least $\frac{49 w_e z_e^*}{144}$.

The expected weight of the stable edges in a solution obtained by Algorithm 2-PHLV is

$$W = \sum_{e \in E} w_e Pr[e \text{ is stable}] \geq \frac{49}{144} \sum_{e \in E} w_e z_e^* \geq \frac{49}{144} OPT.$$

3 Inapproximability

To establish MAXSNP-hardness of MAX-k-EC we give an L-reduction from the well-known Maximum 3-Satisfability problem (MAX-E3-SAT), which cannot be approximated within $\mathbf{\frac{7}{8}} + \epsilon$ for $\epsilon > 0$ unless $\mathbf{P} = \mathbf{NP}$ [7]. MAX-E3-SAT:

We are given a set of Boolean variables $x_1, x_2, ..., x_n$ and a collection of disjunctive clauses $C_1, C_2, ..., C_m$. Every clause has exactly three literals. Find an assignment of Boolean values to the variables which satisfies as many clauses as possible.

Definition 1 (L-reducibility [11]).
Given two optimization problems Π_1 and Π_2, we say we have an L-reduction ($\Pi_1 \leq_L \Pi_2$) with parameters α and β from Π_1 to Π_2, if for some $\alpha, \beta > 0$

- *For each instance I of Π_1 we can compute in polynomial time an instance I' of Π_2;*
- *$OPT(I') \leq \alpha\, OPT(I)$;*
- *Given a solution of value B' to I', we can compute in polynomial time a solution of value B to I such that $|OPT(I) - B| \leq \beta |OPT(I') - B'|$.*

For any given instance I of MAX-E3-SAT we construct a particular instance I' of Max-k-EC, i.e., we define a particular edge-colored graph G. Up to renaming of variables we have four different types of clauses: $(x_i \vee x_j \vee x_k)$, $(x_i \vee x_j \vee \bar{x}_k)$, $(x_i \vee \bar{x}_j \vee \bar{x}_k)$, and $(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$. We start the construction of G from the gadgets given in Figure 1, where the color assigned to each edge is indicated. For the rest of the proof we assume that $\mathcal{C} = R(ed), B(lue), G(reen)$.

For each variable x_i of the MAX-E3-SAT x_i we create a vertex v_i. We construct a gadget for every clause C_i, $i = 1, \ldots, m$. The gadget has seven vertices and nine edges. Six edges join vertices of the gadget and three edges join three vertices of the gadget with three vertices corresponding to variables of C_i.

First type: Assume that all literals are positive i.e., when the clause is $(x_i \vee x_j \vee x_k)$, then we construct the gadgets in Figure 1(a).
Second type: Assume that two literals are positive and one literal is negative. That is the clause is of the form $(x_i \vee x_j \vee \bar{x}_k)$, then we construct the gadgets in Figure 1(b).
Third type: Assume that two literals are negative and one literal is positive. That is the clause is of the form $(x_i \vee \bar{x}_j \vee \bar{x}_k)$, then we construct the gadgets in Figure 1(c).
Forth type: Assume that all literals are negative i.e., when the clause is $(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$, then we construct the gadgets in Fig 1(d).

We note that the constructed graph does not contain any odd cycle and so it is bipartite. Also, for every vertex of the constructed graph the edges that are incident to this vertex are colored with at most three different colors, i.e. the chromatic degree of the graph is equal to three. The following lemma follows directly from the construction of the graph G.

Lemma 8. *The maximum contribution that any gadget can have is exactly 4 and is obtained when at least one of the vertices corresponding to variable has the same color as the edge that connects it with the rest of the gadget. Otherwise the maximum contribution that can be achieved is 3.*

Theorem 2. *MAX-E3-SAT $\leq_L$ MAX-3-EC for $\alpha = \frac{31}{7}$, $\beta = 1$.*

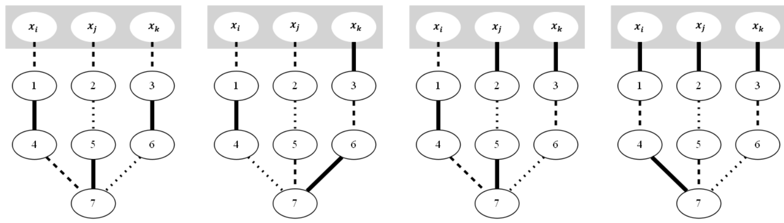

Fig. 1: Gadgets for four types of clauses. Let green, red and blue edges be dashed, thick, and dotted edges, respectively. (a) $(x_i \vee x_j \vee x_k)$. (b) $(x_i \vee x_j \vee \bar{x}_k)$. (c) $(x_i \vee \bar{x}_j \vee \bar{x}_k)$. (d) $(\bar{x}_i \vee \bar{x}_j \vee \bar{x}_k)$.

Proof. First we claim that for an instance I of the MAX-E3-SAT problem there is a truth-assignment that satisfies at least B clauses if and only if there is a vertex coloring with $3m + B$ stable edges. To prove the only-if direction, consider a truth assignment that satisfies at least B clauses. In the graph G, color green all the vertices that correspond to variables that are true and red all the vertices that correspond to false variables. In this way, for each satisfied clause the corresponding gadget in the optimum coloring will have pay-off four. We color all unsatisfied clauses in such a way that each unsatisfied clause has three stable edges. Hence, the total pay-off will be at least $3m + B$. For the opposite direction, suppose that the graph G has a coloring with $3m + B$ stable edges. Since each one of the gadgets has either 3 or 4 stable edges, there must exist at least B gadgets with 4 stable edges. Let us assign the value true to the variables with green corresponding vertices and the value false to the rest of the variables. Notice now that each one of the gadgets with four stable edges corresponds to a satisfied clause. Since such gadgets are at least B, there are at least B clauses that are satisfied and the if direction holds too. Thus, we obtain that $OPT(I') = 3m + OPT(I)$.

Johnson [8] presents the simple algorithm for the maximum satisfiability problem that satisfies at least $\frac{7}{8}$ of the clauses of a MAX-E3-SAT instance, so that $OPT(I) \geq \frac{7}{8}m$.

$$OPT(I') = 3m + OPT(I) \leq 3(\frac{8}{7}OPT(I)) + OPT(I) = \frac{31}{7}OPT(I)$$

Suppose that we have a solution of value $B' = 3m + B$ to the instance I'. Assigning a value of true to variables with green corresponding vertices, and false for the remaining variables we obtain a solution of value B to the instance I. It follows that $|OPT(I) - B| = |OPT(I') - 3m - B| \leq |OPT(I') - B'|$ and we have an L-reduction with parameters $\alpha = \frac{31}{7}$ and $\beta = 1$.

Since L-reductions preserve approximability within constant factors, an L-reduction from a MAX-E3-SAT problem to MAX-k-EC implies that the latter problem is MAXSNP-hard. In particular, MAXSNP-hard problems do not admit a PTAS.

Corollary 1.
MAX-k-EC cannot be approximated better than $\frac{241}{248}$ *unless* ***P=NP****, even though the given graph is bipartite and its edges are colored with at most three different colors.*

Proof. If there is an L-reduction with parameters α and β from maximization problem Π_1 to maximization problem Π_2, and there is an ρ-approximation algorithm for Π_2, then there is an $(1 - \alpha\beta(1 - \rho))$-approximation algorithm for Π_1, see Theorem 16.5 in [11]. Given a ρ'-approximation algorithm for MAX-k-EC, we then have a $\rho = (1 - \frac{31}{7}(1 - \rho'))$-approximation algorithm for the MAX-E3-SAT problem. Hastad [7] proved that no ρ-approximation algorithm exists for MAX-E3-SAT for constant $\rho > \frac{7}{8}$ unless P=NP. By simple calculations, we obtain that no ρ-approximation algorithm exists for MAX-E3-SAT for constant $\rho > \frac{241}{248}$ unless P=NP.

4 Conclusion

For the MAX-k-EC problem we have presented a new approximation algorithm with approximation ratio $\mathbf{\frac{49}{144}}$, improving the previous known ratio $\mathbf{\frac{7}{23}}$. We also have obtained the first upper bound of $\mathbf{\frac{241}{248}}$ for the inapproximability of the MAX-k-EC problem, even for the case when the given graph is bipartite and its edges are colored with at most three different colors. Our approximation algorithm is based on a randomized rounding of a natural LP-relaxation. First, we notice that our algorithm can be derandomized using the method of conditional expectations. Second, note that Theorem 1 provides a lower bound for the integrality gap of the integer linear program $(1) - (4)$. We propose an overview of the results on (in)approximability of the MAX-k-EC problem in Figure 2. Therefore a number of problems remained open for further research.

- Is there a better approximation factor than $\frac{49}{144}$?
- It is interesting to determine what is the maximum possible integrality gap of the integer linear program (ILP) $(1) - (4)$. It is given by Ageev and Kononov [1] that the upper bound of ILP is $\mathbf{\frac{2}{3}}$ by giving a trivial example of triangle with unit edge weights and a unique color for each edge.
- We note that all known approximation algorithms for the MAX-k-EC problem are based on the ILP $(1) - (4)$. It is interesting to design an algorithm using other lower bounding scheme.
- Is there a better upper bound of the inapproximability of the MAX-k-EC problem?

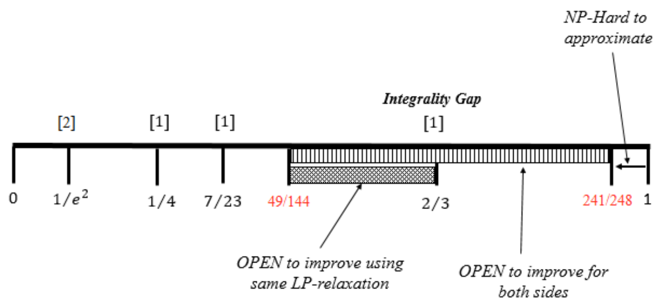

Fig. 2: Overview of results on (in)approximability the MAX-k-EC problem

References

[1] A. Ageev and A. Kononov. "Improved Approximations for the Max k-Colored Clustering Problem". In: *International Workshop on Approximation and Online Algorithms, WAOA 2014* (2015), pp. 1–10.

[2] E. Angel et al. "Clustering on k-Edge-Colored Graphs". In: *MFCS LNCS* 8087 (2013). Ed. by Chatterjee K. and Sgall J., pp. 50–61.

[3] E. Angel et al. "Clustering on k-Edge-Colored Graphs". In: *Discrete Applied Mathematics* 211 (2016), pp. 15–22.

[4] J Bang-Jensen and Gutin G. "Alternating cycles and pathes in edge-coloured multigraphs: a servey". In: *Discrete Mathematics* 165-166 (1997), pp. 39–60.

[5] L. Cai and O.-Y. Leung. "Alternating Path and Coloured Clustering". In: (2018). URL: arXiv:1807.10531.

[6] U. Feige and J. Vondrak. "Approximation algorithm for allocation problem improving the factor of $1 - \frac{1}{e}$". In: *47th Annual IEEE Symposium on Foundations of Computer Science* (2006), pp. 667–676.

[7] J. Hastad. "Some optimal inapproximability results". In: *Journal of the ACM* 48 (4 2001), pp. 798–859.

[8] D. S. Johnson. "Approximation Algorithm For Combinatorial Problems". In: *Journal of Computer and System Sciences* 9.3 (1974), pp. 256–278.

[9] M. Kano and X. Li. "Monochromatic and Heterochromatic Subgraphs in Edge-Colored Graphs - A Survey". In: *Graphs and Combinatorics* 24 (4 2008), pp. 237–263.

[10] J. M. Kleinberg and K. Ligett. "Information-Sharing and Privacy in Social Networks". In: *Games and Economic Behavior* 82 (2013), pp. 702–716.

[11] D. P. Wiliiamson and D. B. Shmoys. *The Design of Approximation Algorithms.* Cambridge University Press, 2011.