Kernelization for Maximum Happy Vertices Problem

Hang Gao[1(✉)] and Wenyu Gao[2]

[1] School of Transportation, Jilin University, Changchun, China
gaohang1998@163.com
[2] School of Information Science, Guangdong University of Finance and Economics, Guangzhou, China
gwy@gdufe.edu.cn

Abstract. The homophyly phenomenon is very common in social networks. The Maximum Happy Vertices (MHV) is a newly proposed problem related to homophyly phenomenon. Given a graph $G = (V, E)$ and a vertex coloring of G, we say that a vertex v is happy if v shares the same color with all its neighbors, and unhappy, otherwise, and that an edge e is happy, if its two endpoints have the same color, and unhappy, otherwise. Given a partial vertex coloring of G with k number of different colors, the k-MHV problem is to color all the remaining vertices such that the number of happy vertices is at least l. We study k-MHV from the parameterized algorithm perspective; we prove that k-MHV has an exponential kernel of $2^{kl+l} + kl + k + l$ on general graph. For planar graph, we get a much better polynomial kernel of $7(kl + l) + k - 10$.

Keywords: Maximum happy vertices · Happy coloring
Parameterized complexity · Kernelization · Planar graph

1 Introduction

Social networks attract more and more researchers. It is believed that homophyly is one of the most basic notions governing the structure of social networks. It is a common sense principle that people are more likely to connect with people they like.

Li and Peng [1] showed that many real networks satisfy exactly the homophyly law. Based on this, Zhang and Li [2] proposed a new problem that, given a network in which some vertices have their attributes unfixed, how to assign attributes to these vertices such that the resulting network reflects the homophyly law in the most degree? Suppose in a company there are many employees which constitutes a friendship network. Some employees have been assigned to work in some departments of the company, while the remaining employees are waiting to be assigned. An employee is happy, if she/he works in the same department with all of her/his friends; otherwise she/he is unhappy. Similarly, a friendship is happy if the two related friends work in the same department;

M. A. Bender et al. (Eds.): LATIN 2018, LNCS 10807, pp. 504–514, 2018.
https://doi.org/10.1007/978-3-319-77404-6_37

otherwise the friendship is unhappy. The goal is to achieve the greatest social benefits, that is, to maximize the number of happy vertices (similarly, happy edges) in the network.

The aforementioned problem can be formally expressed as follows. Consider a vertex-colored graph $G = (V, E)$, an edge is happy if its two endpoints have the same color (otherwise, the edge is unhappy). Similarly, a vertex is happy if it and all its neighbors have the same color (otherwise, the vertex is unhappy). Equivalently, a vertex is happy when all of its incident edges are happy. Let $S \subseteq V$, and let $c : S \to [k]$ be a partial vertex-coloring of G. A full coloring $c' : V \to [k]$ is an extended full coloring of c if $c(v) = c'(v)$ for all $v \in S$. In this paper, we consider the following coloring problems.

Definition 1. *k-Maximum Happy Vertices (k-MHV)*
Instance: A graph $G = (V, E)$, a partial vertex-coloring $c : S \to [k]$, an integer l.
Question: Is there an extended full coloring c' of c such that the number of the happy vertices is at least l?

Definition 2. *k-Maximum Happy Edges (k-MHE)*
Instance: A graph $G = (V, E)$, a partial vertex-coloring $c : S \to [k]$, an integer l.
Question: Is there an extended full coloring c' of c such that the number of the happy edges is at least l?

Though they are newly proposed algorithmic problems, both MHV and MHE problems are natural and fundamental algorithmic problems. The concept of homophyly reflected by it makes it important in social network application. Theoretical analysis of the problems helps to understand the nature of the problems, and hence is very welcome.

Zhang and Li [2] proved that for every $k \geq 3$, the problems k-MHE and k-MHV are NP-complete. However, when $k = 2$, they gave algorithms running in time $O(min\{n^{2/3}m; m^{3/2}\})$ and $O(mn^7 logn)$ for 2-MHE and 2-MHV, respectively. Towards this end, the authors used max-flow algorithms (2-MHE) and minimization of submodular functions (2-MHV). Moreover, the authors presented approximation algorithms with approximation ratios 1/2 and $max\{1/k, \Omega(\Delta^{-3})\}$ for k-MHE and k-MHV, respectively, where Δ is the maximum degree of the graph. Later on, Zhang et al. [3] gave improved algorithms with approximation ratios 0.8535 and $1/(\Delta+1)$ for k-MHE and k-MHV, respectively.

Aravind et al. [4] proved that both k-MHE and k-MHV are solvable in polynomial time for trees, and they proposed an interesting problem of studying the hardness of the k-MHV problem for planar graphs. Furthermore, Aravind et al. [5] proved an polynomial kernel for k-MHE problem from the parameterized algorithm perspective, they also proposed an open question that whether k-MHV admits a polynomial kernel.

Though the polynomial kernel for k-MHE problem proved by Aravind et al. [5] seems quite well, there is no kernelization algorithm for k-MHV problem currently, which inspires us to conduct a further study for k-MHV problem from the parameterized algorithm perspective.

2 Preliminaries

Throughout, we consider graphs that are finite, undirected and simple. For a graph $G = (V, E)$, let $V(G)$ denote its vertex set and $E(G)$ its set of edges. For $S \subseteq V$, the subgraph induced by S, denote $G[S]$, is the subgraph of G with vertex set S and edge set $\{(u, v)|u, v \in S, (u, v) \in E\}$. For each vertex $v \in V(G)$, let $N(v)$ be the set of vertices adjacent to v in G, i.e., $N(v) = \{u \in V|(u, v) \in E\}$, let $N_S(v)$ denote the set of vertices adjacent to v in S, where S is a subset of $V(G)$, i.e., $N_S(v) = \{x|x \in N(v) \text{ and } x \in S\}$.

The degree $deg(v)$ of v is the size of $N(v)$. A vertex of degree exactly (at most, at least) d is called a d-vertex ((d)-vertex, (d)-vertex). Let $\Delta(G)$ denote the maximum degree over all vertices in G.

If $u, v \in V$ are two vertices of graph G, the operation of contracting u and v is to combine u and v into a new vertex w, which becomes adjacent to all the former neighbors of u and of v.

A planar graph is a graph that can be embedded in the plane. For a given planar graph $G = (V, E)$, the following condition holds for $|V| \geq 3$, $|E| \leq 3|V| - 6$. The subdivision of an edge $e = (u, v)$ with endpoints u and v yields a graph containing one new vertex w, and with an edge set replacing e by two new edges, (w, u) and (w, v). For a planar graph, the subdivision of arbitrary edge will not change the planarity of the graph.

The theory of parameterized computation and complexity mainly considers decision problems (i.e., problems whose instances only require a yes/no answer). A parameterized problem Q is a decision problem (i.e., a language) that is a subset of $\Sigma^* \times N$, where Σ is a fixed alphabet and N is the set of all nonnegative integers. Thus, each element of Q is of the form (x, k), where the second component, i.e., the integer k, is the parameter. We say that an algorithm A solves the parameterized problem Q if on each input (x, k), the algorithm A can determine whether (x, k) is a yes-instance of Q (i.e., whether (x, k) is an element of Q). We call the algorithm A a parameterized algorithm if its computational complexity is measured in terms of both the input length $|x|$ and the parameter value k. The parameterized problem Q is fixed parameter tractable if it can be solved by a parameterized algorithm of running time bounded by $f(k)|x|^c$, where f is a recursive function and c is a constant independent of both k and $|x|$.

Kernelization is one of the most important techniques used in the development of efficient parameterized algorithm. Let Q be a parameterized problem and (x, k) be an instance of Q. An algorithm K is called a kernelization algorithm for Q if K satisfies the following conditions: (1) K transforms (x, k) into the reduced instance (x', k') in polynomial time; (2) (x, k) is a yes-instance of Q if and only if (x', k') is a yes-instance of Q; and (3) $|x'| \leq g(k)$ and $k' \leq k$, where $g(k)$ is a computable function. Correspondingly, the problem Q is called kernelizable and the reduced instance (x', k') is called a kernel. In particular, Q is said to admit a polynomial kernel if $g(k)$ is a polynomial function on k. One of the most important theorems of parameterized computation is that a parameterized problem is fixed parameter tractable if and only if it is kernelizable [6].

The reduced kernel is not only helpful for parameterized algorithm, but also helpful for approximation algorithm. For more on parameterized complexity, we refer the interested reader to [7,8].

3 Kernelization for *k*-MHV Problem

According to the description of k-MHV, the difficulty is that since the original graph is a partial-colored graph, it is very hard to determine whether an colored or uncolored vertex is a happy vertex of the final optimal solution. If the graph is dense enough (for instance, complete graph), only two pre-colored vertices with different colors will make all the rest vertices be unhappy vertices. On the contrary, given a sparse graph, it is easier to process.

From the perspective of parameterized algorithm, kernelization is to reduce the original graph G to a "small" kernel. In the following, we prove k-MHV has a exponential kernel of $2^{kl+l} + kl + k + l$, thus k-MHV is fixed parameter tractable. Furthermore, in planar graph, k-MHV has a much better kernel of $7(kl + l) + k - 10$. Our strategy to obtain the kernels consists of three parts: firstly, we partition the vertices of the partial-colored graph into six disjoint sets, secondly, we introduce four reduction rules to reduce the graph; thirdly, we introduce a strategy to analyze the size of each disjoint set of vertices.

3.1 Partition of Vertices

Given a partial-colored graph $G = (V, E)$, we can partition the vertices of G into six disjoint sets.

(1) Set of colored happy vertices, denoted by V_1, i.e., $v \in V_1$ if v is colored and all the vertices in $N(v)$ have been colored with the same color as that of v.
(2) Set of colored unhappy vertices, denoted by V_2, i.e., $v \in V_2$ if v is colored and destined to be unhappy. That is to say, there is at least one vertex $u \in N(v)$, which has been colored with the color different from that of v.
(3) Set of colored potential happy vertices, denoted by V_3, i.e., $v \in V_3$ if v is colored and v has the same color as all of its colored neighbors, but v also has some uncolored neighbors.
(4) Set of uncolored unhappy vertices, denoted by V_4, i.e., $v \in V_4$ if v is uncolored and v has at least two colored neighbors with different colors.
(5) Set of uncolored potential happy vertices, denoted by V_5, i.e., $v \in V_5$ if v is uncolored and all the colored neighbors of v have the same color, but v also has some uncolored neighbors.
(6) Set of uncolored free vertices, denoted by V_6, i.e., $v \in V_6$ if v is uncolored and v has no colored neighbors.

The partition of the vertices is shown in Fig. 1. Black vertices are colored vertices; the numbers beside those colored vertices represent their color numbers. White vertices are uncolored vertices.

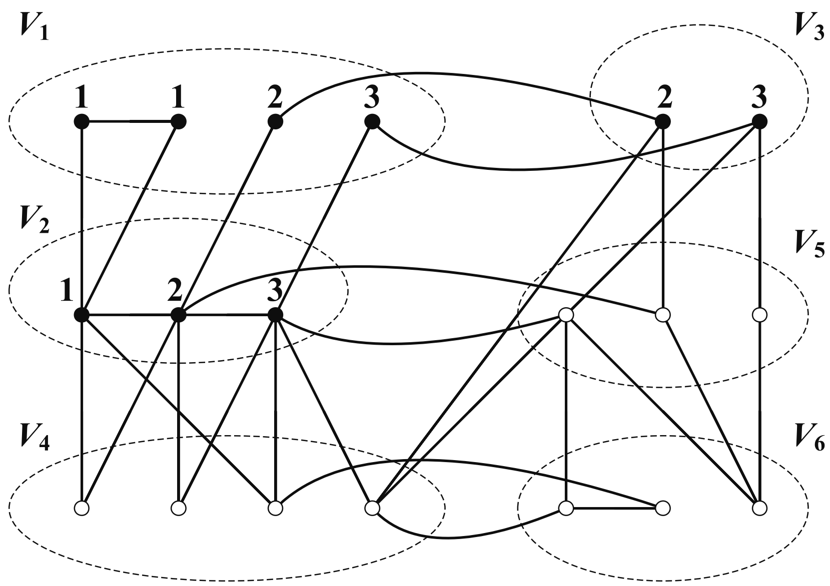


Fig. 1. Partition of vertices of G. V_1 is the set of colored happy vertices, V_2 is the set of colored unhappy vertices, V_3 is the set of colored potential happy vertices, V_4 is the set of uncolored unhappy vertices, V_5 is the set of uncolored potential happy vertices, V_6 is the set of uncolored free vertices.

3.2 Reduction Rules

Now we present the following four reduction rules.

Rule 1: Remove all the happy edges, because the happy edges will never further affect the coloring of the remaining vertices. By removing the happy edges, all the colored happy vertices (vertices in V_1) in graph G become isolated vertices. Then remove all the isolated vertices and decrease l by the number of removed vertices. By doing so, the set of V_1 no longer exists in the reduced graph.

Rule 2: Considering the colored unhappy vertices in V_2, contract all the vertices with the same color into a single vertex and remove all but one of the parallel edges.

Proof. According to the partition of the vertices, vertices in V_2 are unhappy vertices; they will never further contribute to the value of the optimal solution. How many colored vertices are adjacent to an uncolored vertex is not serious, but what colored vertices are adjacent to an uncolored vertex is the key problem. Moreover, after applying of Rule 1, all the vertices with the same color in V_2 forms an independent set. Therefore, we can contract all the vertices with the same color into a single vertex and remove the parallel edges safely.

Rule 3: If there are some vertices in V_4 only adjacent to vertices in V_2, remove them from G.

Proof. The vertices in V_4 only adjacent to vertices in V_2 will never affect the colored vertices in V_3, the uncolored vertices in V_5, and the free vertices in V_6. So they can be removed in advance.

Rule 4: For the rest of uncolored unhappy vertices in V_4, they are all adjacent to vertices in V_3 or V_5, or vertices in V_6. That is to say, they are adjacent to at least one vertex in $V_3 \cup V_5 \cup V_6$. If there are two vertices $u, v \in V_4$, and $N_{V_3 \cup V_5 \cup V_6}(u) \subseteq N_{V_3 \cup V_5 \cup V_6}(v)$, then remove vertex u.

Proof. Vertices in V_4 are destined to be unhappy vertices, so in the optimal solution, they can be colored with the same color as one of its neighbors in $V_3 \cup V_5 \cup V_6$ safely. Suppose in the optimal solution, vertex u is colored with color c_1, and vertex v is colored with color c_2, thus all the neighbors of u are unhappy vertices. So it is safe to change the color of u from c_1 to c_2, which will not decrease the number of happy vertices. Thus if there are two vertices u and v in V_4 such that $N_{V_3 \cup V_5 \cup V_6}(u) \subseteq N_{V_3 \cup V_5 \cup V_6}(v)$, the best choice is to color them with the same color. Because u and v will be colored with the same color in the end, and $N_{V_3 \cup V_5 \cup V_6}(u) \subseteq N_{V_3 \cup V_5 \cup V_6}(v)$, then u can be removed in advance.

Therefore, after applying of Rule 4, for each pair of vertices $u, v \in V_4$, the following two forms are satisfied.

$$\begin{aligned} N_{V_3 \cup V_5 \cup V_6}(u) \not\subseteq N_{V_3 \cup V_5 \cup V_6}(v) \\ N_{V_3 \cup V_5 \cup V_6}(u) \not\supseteq N_{V_3 \cup V_5 \cup V_6}(v) \end{aligned} \tag{1}$$

The reduced graph G' by applying Rule 1 to Rule 4 is shown in Fig. 2. It is worth noting that the happy edges and the colored happy vertices have been removed from the graph.

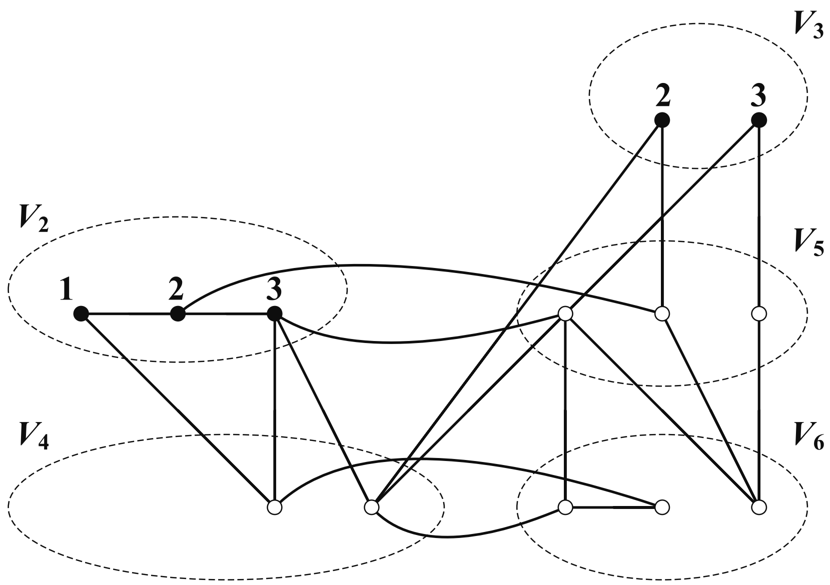


Fig. 2. The reduced graph G'. Black vertices are colored vertices; the numbers beside these colored vertices are the colors of them. White vertices are uncolored vertices.

3.3 Analyzing Size of Vertex Set

After applying the aforementioned four reduction rules, we are ready to conclude the kernel of k-MHV.

Theorem 3. *The problem k-MHV admits a kernel of* $2^{(k+l)} + kl + k + l$ *vertices. Thus, k-MHV is fixed parameter tractable.*

Proof. Let (G', k, l) be a reduced instance of k-MHV. We claim that if G' has more than $2^{(k+l)} + kl + k + l$ vertices, then we have YES-instance. The proof follows by the claims below.

Claim 1.1. The number of vertices in V_1 is 0, i.e.,

$$|V_1| = 0 \tag{2}$$

It is obvious according to Rule 1.

Claim 1.2. The number of vertices in V_2 is at most k, i.e.,

$$|V_2| \leq k \tag{3}$$

Proof. By applying Rule 2, all the vertices with the same color are contracted into a single vertex, so there are at most k vertices with different colors in V_2.

Claim 1.3. The number of free vertices in V_6 is at most $l - 1$, i.e.,

$$|V_6| < l \tag{4}$$

Proof. For the uncolored free vertices in V_6, if there are at least l uncolored free vertices, we can conclude that there is a solution that contains at least l happy vertices. Because in that case, we can color all the uncolored vertices of graph G' with the same color, which makes all the uncolored free vertices in V_6 become happy vertices. So $|V_6| < l$.

Claim 1.4. The number of vertices in $V_3 \cup V_5$ is at most $kl - 1$, i.e.,

$$|V_3 \cup V_5| < kl \tag{5}$$

Proof. For arbitrary uncolored potential happy vertex $u, u \in V_5$, if the color number of its colored neighbors is c_1, then we call c_1 as the potential color number of vertex u. If there are at least kl potential happy vertices (colored or uncolored) in $V_3 \cup V_5$, we can choose at least $kl/k = l$ vertices whose color number or potential color number are the same, because there are at most k different color numbers. Then we can make these l vertices become happy vertices. This is feasible because we can color all of them and their neighbors with their original color number or potential color number. Thus, $|V_3 \cup V_5| < kl$.

Claim 1.5. The number of vertices in V_4 is at most 2^{kl+l}, i.e.,

$$|V_4| < 2^{kl+l} \tag{6}$$

Proof. By applying Rules 3 and 4, each vertex in V_4 is adjacent to at least one vertex in $V_3 \cup V_5 \cup V_6$, and for each pair of vertices $u, v \in V_4$, $N_{V_3 \cup V_5 \cup V_6}(u) \nsubseteq N_{V_3 \cup V_5 \cup V_6}(v)$, and $N_{V_3 \cup V_5 \cup V_6}(u) \nsupseteq N_{V_3 \cup V_5 \cup V_6}(v)$. According to Claims 1.3 and 1.4, $|V_3 \cup V_5 \cup V_6| < kl + l$. Because the set of $V_3 \cup V_5 \cup V_6$ has at most $\binom{kl+l}{(kl+l)/2}$ number of subsets that do not contain each other. Thus, $|V_4| < \binom{kl+l}{(kl+l)/2} < 2^{kl+l}$.

Therefore, the number of vertices after applying reduction rules is,

$$|V_1 \cup V_2 \cup V_3 \cup V_4 \cup V_5 \cup V_6| < 0 + k + kl + 2^{kl+l} + l = 2^{kl+l} + kl + k + l \quad (7)$$

This is an exponential kernel of k-MHV problem. Furthermore, all of the reduction rules can be implemented to run in polynomial time. Thus, the claimed kernel follows. □

According to Rule 1 to Rule 4 and Theorem 3, we can easily design a kernelization algorithm for k-MHV problem, the kernelization algorithm just need to include the four reduction rules.

3.4 A k-approximation Algorithm

It is worth noting that the strategy used to bound $|V_3 \cup V_5|$ in proof of Theorem 3 (Claim 1.4) can be extended to an approximation algorithm. The main idea is that we can choose at least $|V_3 \cup V_5|/k$ vertices whose color number or potential color number are the same, and make these vertices become happy vertices by coloring all of them and their neighbors with their original color number or potential color number.

So we can change at least $|V_3 \cup V_5|/k$ vertices into happy vertices every time. If $V_3 \cup V_5$ is null, we only need to color all the vertices in V_6 and all their neighbors in V_4 with the same color, which lead to the end of the approximation algorithm.

Given a partial-colored graph G, the vertices in V_2 and V_4 will never become happy vertices, the vertices in V_1 will not affect the rest coloring procedure. So in the best case, we can assume that the vertices in V_3, V_5, and V_6 will all become happy vertices in the end. But by our approximation algorithm, we can make at least $|V_3 \cup V_5|/k$ vertices into happy vertices every time, till the end. Thus this is a k-approximation algorithm.

4 Maximum Happy Vertices on Planar Graph

It is obvious that the graph is more dense, there are fewer happy vertices. Thus, it is natural to ask what the parameterized complexity of k-MHV on planar graph is. For a planar graph $G = (V, E)$, $|E| \leq 3|V| - 6$.

Zhang and Li [2] proved the NP-hardness of k-MHE by reducing multiway cut problem to k-MHE, and they also proved the NP-hardness of k-MHV by reducing k-MHE to k-MHV. Dahlhaus et al. [9] proved that the multiway cut problem is NP-hard even on planar graphs. Thus, it is obvious that k-MHV on planar graphs is also NP-hard. To bound the kernel of k-MHV on planar graph, we use a method proposed by Wang et al. [10] to count the vertices on planar graph.

The proof of Theorem 3 shows that the cardinalities of V_3, V_5, and V_6 are bounded to polynomial functions of k or l, but the cardinality of V_4 is related to an exponential function of k and l. Thus, our main idea is to bound the cardinality of V_4 by taking advantage of the properties of planar graph.

Let's consider the subgraph containing Vertex set V_4 and $V_3 \cup V_5 \cup V_6$, and the edges between them. It is a bipartite graph, we denote it by $B = (V_4 \cup V_{3-5-6}, E)$, where $V_{3-5-6} = V_3 \cup V_5 \cup V_6$. At the beginning, if the given graph G is a planar graph, during the reduction of graph G, applying reduction Rule 1 will not change the planarity of graph G, applying reduction Rule 2 may change the planarity of graph G, but it will not affect the planarity of the induced subgraph $G[V_4 \cup V_3 \cup V_5 \cup V_6]$. The aforementioned bipartite graph $B = (V_4 \cup V_{3-5-6}, E)$ is a subgraph of $G[V_4 \cup V_3 \cup V_5 \cup V_6]$, so $B = (V_4 \cup V_{3-5-6}, E)$ is a bipartite planar graph. Clearly, applying of Rules 3 and 4 will remove some vertices in V_4, which will not change the planarity of $B = (V_4 \cup V_{3-5-6}, E)$.

Therefore, given a planar graph G, after applying reduction rules, the reduced graph G' may be a non-planar graph, but the subgraph $B = (V_4 \cup V_{3-5-6}, E)$ of G' is a bipartite planar graph.

Based on this condition, we arrive at the kernel size of k-MHV on planar graph.

Theorem 4. *The problem k-MHV on planar graph admits a linear kernel of* $7(kl + l) + k - 10$ *vertices.*

Proof. In Theorem 3, we prove that $|V_2| \leq k$, $|V_3 \cup V_5| < kl$, $|V_6| < l$ after using four reduction rules. Now let us count V_4 on the condition of the subgraph $B = (V_4 \cup V_{3-5-6}, E)$ is a planar graph.

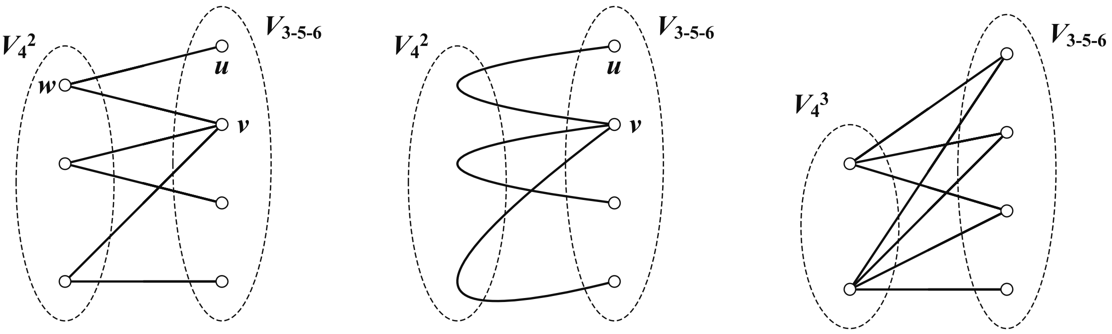
Firstly, we partition V_4 into three disjoint sets, V_4^1 denotes the set of vertices which have only one neighbor in V_{3-5-6}, V_4^2 denotes the set of vertices which have exact two neighbors in V_{3-5-6}, and V_4^3 denotes the set of vertices which have at least three neighbors in V_{3-5-6}.

Secondly, let's bound the number of vertices in V_4^1. Because of each vertex in V_4 is adjacent to at least one vertex of V_{3-5-6}, and for each pair of vertices $u, v \in V_4$, $N_{V_3 \cup V_5 \cup V_6}(u) \nsubseteq N_{V_3 \cup V_5 \cup V_6}(v)$, and $N_{V_3 \cup V_5 \cup V_6}(u) \nsupseteq N_{V_3 \cup V_5 \cup V_6}(v)$. Thus, we have

$$|V_4^1| \leq |V_{3-5-6}| = kl + l \tag{8}$$

Thirdly, let's bound the number of vertices in V_4^2. Each vertex in V_4^2 is adjacent to different pair of vertices in V_{3-5-6}, and $B = (V_4 \cup V_{3-5-6}, E)$ is a bipartite planar graph. Let's consider the subgraph of B formed by V_4^2, V_{3-5-6}, and the edges between V_4^2 and V_{3-5-6}. We denote it by B_2, a example is shown in Fig. 3(a). It is clear that this subgraph B_2 is also a bipartite planar graph, and for each pair of vertices $u, v \in V_{3-5-6}$, there is at most one vertex $w \in V_4^2$ adjacent to both u and v (because $N_{V_3 \cup V_5 \cup V_6}(u) \nsubseteq N_{V_3 \cup V_5 \cup V_6}(v)$, and $N_{V_3 \cup V_5 \cup V_6}(u) \nsupseteq N_{V_3 \cup V_5 \cup V_6}(v)$). Assuming we replace every vertex $w \in V_4^2$ and its two incident edges (w, u) and (w, v) by a single edge (u, v) (this is a reverse operation of subdivision), let B_2' denote the new graph by replacing operation, a example is shown in Fig. 3(b). Clearly, B_2' is a planar graph whose vertex set is V_{3-5-6}. Moreover, there is a one-to-one correspondence between the edge set of B_2' and the replaced vertices in V_4^2. Since every planar graph with $|V|$ vertices and $|E|$ edges satisfies $|E| \leq 3|V| - 6$, therefore,

$$|V_4^2| \leq 3|V_{2-4-5}| - 6 = 3(kl + l) - 6 \tag{9}$$



(a) B_2 (b) B_2' (c) B_3

Fig. 3. Examples of bipartite planar graphs, (a) B_2 is a bipartite planar graph formed by V_4^2, V_{3-5-6}, and the edges between V_4^2 and V_{3-5-6}, (b) B_2' is transformed from B_2 by replacing every vertex $w, w \in V_4^2$ and its two incident edges (w, u) and (w, v) by a single edge (u, v), (c) B_3 is a bipartite planar graph formed by V_4^3, V_{3-5-6}, and the edges between V_4^3 and V_{3-5-6}.

Fourthly, let's bound the number of vertices in V_4^3. Each vertex in V_4^3 is adjacent to at least three vertices in V_{3-5-6}, and $B = (V_4 \cup V_{3-5-6}, E)$ is a bipartite planar graph. Considering the subgraph of B formed by V_4^3, V_{3-5-6}, and the edges between V_4^3 and V_{3-5-6}, denoted by B_3, shown in Fig. 3(c). Clearly, this subgraph is also a bipartite planar graph, which is a triangle-free planar graph. Since triangle-free planar graph with $|V|$ vertices and $|E|$ edges satisfies $|E| \leq 2|V| - 4$, let's consider the edges between V_4^3 and V_{3-5-6}, we have

$$\begin{aligned} &3|V_4^3| \leq |E| \leq 2(|V_4^3| + |V_{3-5-6}|) - 4 \\ &\Rightarrow |V_4^3| \leq 2|V_{3-5-6}| - 4 \\ &\Rightarrow |V_4^3| \leq 2(kl + l) - 4 \end{aligned} \tag{10}$$

Therefore,

$$\begin{aligned} |V_4| &= |V_4^1 \cup V_4^2 \cup V_4^3| \\ &= |V_4^1| + |V_4^2| + |V_4^3| \\ &\leq 6(kl + l) - 10 \end{aligned} \tag{11}$$

After reduction, the number of vertices is,

$$|V_1 \cup V_2 \cup V_3 \cup V_4 \cup V_5 \cup V_6| < 0 + k + kl + l + 6(kl + l) - 10 = 7(kl + l) + k - 10 \tag{12}$$

So k-MHV on planar graph admits a kernel of $7(kl + l) + k - 10$. $\square$

5 Conclusions

The k-MHV problem is a natural graph coloring problem arising in the homophyly phenomenon of networks. By vertex partition, we prove an exponential kernel of $2^{kl+l} + kl + k + l$ for this problem. For planar graph, we can achieve a much better polynomial kernel of $7(kl + l) + k - 10$.

In general, many NP-hard problems are less difficult in some special graphs, and studying of NP-hard problems in special graphs can be very helpful for solving these problems in general graphs. Considering MHV is a newly proposed problem, it is necessary to investigate the complexity of MHV in more special classes of graphs.

Acknowledgments. We would like to thank the anonymous reviewers for their detailed reviews and suggestions.

References

1. Li, A., Peng, P.: The small-community phenomenon in networks. Math. Struct. Comput. Sci. **22**(3), 373–407 (2012)
2. Zhang, P., Li, A.: Algorithmic aspects of homophyly of networks. Theor. Comput. Sci. **593**(C), 117–131 (2015)
3. Zhang, P., Jiang, T., Li, A.: Improved approximation algorithms for the maximum happy vertices and edges problems. In: Xu, D., Du, D., Du, D. (eds.) COCOON 2015. LNCS, vol. 9198, pp. 159–170. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21398-9_13
4. Aravind, N.R., Kalyanasundaram, S., Kare, A.S.: Linear time algorithms for happy vertex coloring problems for trees. In: Mäkinen, V., Puglisi, S.J., Salmela, L. (eds.) IWOCA 2016. LNCS, vol. 9843, pp. 281–292. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-44543-4_22
5. Aravind, N.R., Kalyanasundaram, S., Kare, A.S., Lauri, J.: Algorithms and hardness results for happy coloring problems. CoRR abs/1705.08282 (2017)
6. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford University Press, Oxford (2002)
7. Chen, J.E.: Parameterized computation and complexity: a new approach dealing with np-hardness. J. Comput. Sci. Technol. **20**(1), 18–37 (2005)
8. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer International Publishing, Heidelberg (2015)
9. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. SIAM J. Comput. **23**(4), 864–894 (1994)
10. Wang, J., Yang, Y., Guo, J., Chen, J.: Planar graph vertex partition for linear problem kernels. J. Comput. Syst. Sci. **79**(5), 609–621 (2013)