An Efficient Fixed-Parameter Algorithm for the 2-Plex Bipartition Problem*

Li-Hsuan Chen[†1], **Sun-Yuan Hsieh**[2], **Ling-Ju Hung**[‡3], **and Peter Rossmanith**[4]

1 **Department of Computer Science and Information Engineering National Cheng Kung University, Tainan, Taiwan**
`clh100p@cs.ccu.edu.tw`

2 **Department of Computer Science and Information Engineering National Cheng Kung University, Tainan, Taiwan**
`hsiehsy@mail.ncku.edu.tw`

3 **Department of Computer Science and Information Engineering National Cheng Kung University, Tainan, Taiwan**
`hunglc@cs.ccu.edu.tw`

4 **Department of Computer Science, RWTH Aachen, Aachen, Germany**
`rossmani@cs.rwth-aachen.de`

Abstract

Given a graph $G = (V, E)$, an *s-plex* $S \subseteq V$ is a vertex subset such that for $v \in S$ the degree of v in $G[S]$ is at least $|S| - s$. An *s-plex bipartition* $\mathcal{P} = (V_1, V_2)$ is a bipartition of $G = (V, E)$, $V = V_1 \uplus V_2$, satisfying that both V_1 and V_2 are s-plexes. Given an instance $G = (V, E)$ and a parameter k, the s-Plex Bipartition problem asks whether there exists an s-plex bipartition of G such that $\min\{|V_1|, |V_2|\} \leq k$. The s-Plex Bipartition problem is NP-complete. However, it is still open whether this problem is fixed-parameter tractable. In this paper, we give a fixed-parameter algorithm for 2-Plex Bipartition running in time $O^*(2.4143^k)$. A graph $G = (V, E)$ is called *defective* (p, d)*-colorable* if it admits a vertex coloring with p colors such that each color class in G induces a subgraph of maximum degree at most d. A graph G admits an s-plex bipartition if and only if the complement graph of G, $\bar{G}$, admits a defective $(2, s - 1)$-coloring such that one of the two color classes is of size at most k. By applying our fixed-parameter algorithm as a subroutine, one can find a defective $(2, 1)$-coloring with one of the two colors of minimum cardinality for a given graph in $O^*(1.5539^n)$ time where n is the number of vertices in the input graph.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases 2-plex, 2-plex bipartition, bounded-degree-1 set bipartition, defective $(2, 1)$-coloring

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.20

* Parts of this work were supported by the Ministry of Science and Technology of Taiwan under grants MOST 105–2221–E–006–164–MY3 and MOST 103–2221–E–006–135–MY3.

† Li-Hsuan Chen is supported by the Ministry of Science and Technology of Taiwan under grant MOST 106–2811–E–006–008

‡ Ling-Ju Hung is supported by the Ministry of Science and Technology of Taiwan under grant MOST 106–2811–E–006–055.

1 Introduction

Given a graph $G = (V, E)$, an *s-plex* $S \subseteq V$ is a vertex subset such that for $v \in S$ the degree of v in $G[S]$ is at least $|S| - s$. The notion of s-plexes is a degree relaxed variant of cliques and was defined to study the cohesiveness of subgroups in social networks [24]. It is NP-hard to find an s-plex of maximum cardinality in general graphs [3, 19]. Variants algorithms (*e.g.,* [2, 3, 4, 6, 7]) were designed for finding an s-plex of maximum cardinality in a given graph. Note that the maximum 1-plex problem is exactly the maximum clique problem. The maximum s-plex is trivial when $s = |V|$. An *s-plex bipartition* $\mathcal{P} = (V_1, V_2)$ is a bipartition of $G = (V, E)$, $V = V_1 \uplus V_2$, satisfying that both V_1 and V_2 are s-plexes. Given an instance $G = (V, E)$ and a parameter k, the s-Plex Bipartition problem asks whether there exists an s-plex bipartition of G such that $\min\{|V_1|, |V_2|\} \leq k$.

s-Plex Bipartition
Instance: A graph $G = (V, E)$
Parameter: An integer $k \geq 0$
Question: Does there exist a bipartition $\mathcal{P} = (V_1, V_2)$ such that both V_1 and V_2 are s-plexes and $\min\{|V_1|, |V_2|\} \leq k$?

Graph coloring is often used to model scheduling problems [18, 20]. Given a set of jobs J, one can construct a conflict graph $G = (V, E)$ where $V = J$ and for two jobs $u, v \in V$ having schedule conflicts, there is an edge $uv \in E$. We say that G admits a *proper p*-coloring if vertices in G can be colored with p colors and no two adjacent vertices in G are in the same color class. If vertices (jobs) are in the same color class, then those jobs can be done simultaneously without any conflict. However, the ordinary coloring may be too restricted to model a real scheduling problem in which jobs could tolerate some threshold of conflicts. This gives a more general coloring problem called *defective (p, d)-coloring* introduced in [1, 8, 16]. A vertex subset $S \subseteq V$ is called a *bounded-degree-d set* if the maximum degree of $G[S]$ is at most d. A graph $G = (V, E)$ is called *(p, d)-colorable* if it admits a vertex coloring with p colors such that each color class in G is a bounded-degree-d set. Here d means *defects* and the threshold of conflicts.

Defective (p, d)-Coloring
Input: A graph $G = (V, E)$
Question: Does there exist a (p, d)-coloring of G?

The notation $\chi_d(G)$ called the *defective chromatic number* of G is to denote the minimum p such that G is (p, d)-colorable and $\chi_0(G)$ is the usual chromatic number of the graph G. We see that a defective $(p, 0)$-coloring is a proper coloring.

The s-Plex Bipartition problem is important because it is related to the Defective (p, d)-Coloring. It is not hard to see that a graph G admits an s-plex bipartition if and only if the complement graph of G, $\bar{G}$, is defective $(2, s-1)$-colorable. The problem to determine whether an input graph is defective $(2, 0)$-colorable is equivalent to the recognition of bipartite graphs and can be done in linear time. Surprisingly, the Defective $(2, 1)$-coloring problem is NP-complete for general graphs [9] and even for planar graphs [10] and for graphs of maximum degree 4 [10]. This generalizes that Defective $(2, d)$-coloring is NP-complete for any $d \geq 1$ in general graphs and planar graphs. Moreover, the Defective (p, d)-Coloring is NP-complete for all $p \geq 3$ and $d \geq 0$ in general graphs [10]. To determine whether a planar graph is defective $(3, 1)$-colorable is also NP-complete [10]. It was proved that for any constant d, there exists an $\epsilon > 0$ such that $\chi_d(G)$ can not be approximated within a factor of n^{ϵ} unless P=NP [10].

Lovász [21] showed that for any positive integer p, any graph $G = (V, E)$ of maximum degree $\Delta(G)$ admits a defective $(p, \lfloor \Delta(G)/p \rfloor)$-coloring and the coloring can be found in time $O(\Delta(G) \cdot |E|)$. The defective chromatic number of planar graphs has been well-studied in [8, 14, 15, 17, 23, 25]. It was proved that any planar graph admits a defective $(3, 2)$-coloring and can be found in $O(n^2)$ time [8]. Poh [23] and Goddard [15] showed that any planar graph admits a special defective $(3, 2)$-coloring in which each color class is the disjoint union of paths.

A problem is *fixed-parameter tractable* (FPT) if given any instance of size n and a positive integer k, one can give algorithms to solve it in time $f(k) \cdot poly(n)$ where $f(k)$ is a computable function only depending on k. Those algorithms are called *fixed-parameter algorithms*. There are many results about fixed-parameter algorithms introduced in [11, 12].

A fixed-parameter algorithm based on *branch-and-reduce* strategy consists of a collection of *reduction rules* and *branching rules*. Given a problem instance (G, k) with the parameter k, reduction rules are used to obtain a smaller problem instance (G', k') in polynomial time such that $|G'| < |G|$ or $k' < k$. The branching rules are used to recursively solve the smaller instances of the problem with smaller parameter. We analyze each branching rule and use the worst-case time complexity over all branching rules as an upper bound of the running time. Search trees are often used to illustrate the execution of a branching algorithm. The root of a search tree represents the input of the problem, every child of the root represents a smaller instance reached by applying a branching rule associated with the instance of the root. One can recursively assign a child to a node in the search tree when applying a branching rule. Notice that we do not assign a child to a node when applying a reduction rule. The running time of a branching algorithm is usually measured by the maximum number of leaves in its corresponding search tree.

Let b be any branching rule. When rule b is applied, the current instance (G, k) is branched into $r \geq 2$ instances (G_i, k_i) where $|G_i| \leq |G|$ and $k_i = k - t_i$. Notice that fixed-parameter algorithms return "No" when the parameter $k \leq 0$. We call $\mathbf{b} = (t_1, t_2, \ldots, t_r)$ the *branching vector* of branching rule b. This can be formulated in a linear recurrence

$$T(k) \leq T(k - t_1) + T(k - t_2) + \cdots + T(k - t_r)$$

where $T(k)$ is the number of leaves in the search tree depending on the parameter k. The running time of the branching algorithm using only branching rule b is $O(poly(n) \cdot T(k)) = O^*(c^k)$[1], where c is the unique positive real root of $x^k - x^{k-t_1} - x^{k-t_2} - \cdots - x^{k-t_r} = 0$ [13]. The number c is called the *branching number* of the branching vector $(t_1, t_2, \ldots, t_r)$.

To the best of our knowledge, whether s-Plex Bipartition admits a fixed-parameter algorithm is an open problem. We first give a simple fixed-parameter algorithm to solve the s-Plex Bipartition problem runs in time $O^*((s + 1)^k)$ by reducing the problem to Minimum Ones $(s + 1)$-SAT.

The following Bounded-Degree-d Set Bipartition problem is equivalent to the Defective $(2, d)$-Coloring problem. Notice that G admits a s-plex bipartition if and only if $\bar{G}$ has a bounded-degree-$(s - 1)$ set bipartition. Moreover, G has a bounded-degree-d set bipartition if and only if G is defective $(2, d)$-colorable and there exists one color class having at most k vertices.

[1] For functions f and g we write $f(k, n) = O^*(g(k))$ if $f(k, n) = O(g(k) \cdot poly(n))$, where $poly(n)$ is a polynomial.

Bounded-Degree-d Set Bipartition (BD-dSB)
Instance: A graph $G = (V, E)$
Parameter: An integer $k \geq 0$
Question: Does there exist a vertex bipartition (B, W) with $V = B \uplus W$ such that both B and W are bounded-degree-d sets and $|B| \leq k$?

We first show that the BD d-SB problem can be reduced to the following Minimum Ones $(d + 2)$-SAT problem and can be solved in time $O^*((d + 2)^k)$.

Minimum Ones $(d + 2)$-SAT
Instance: A $(d + 2)$-CNF formula F.
Parameter: An integer $k \geq 0$
Question: Does there exist a 0/1 satisfying assignment for F such that the number of ones is at most k?

The BD-dSB can be formulated with $(d + 2)$-CNF formula F as follows.

- For each vertex v in the input graph G, create a variable x_v.
- For each $(d + 2)$ star S_{d+2} in G with center u and $(d + 1)$ leaves $v_1, v_2, \ldots, v_{d+1}$ being $(d + 1)$ neighbors of u in G, create two clauses $(x_u \vee x_{v_1} \vee \cdots \vee x_{v_d} \vee x_{v_{d+1}})$ and $(\bar{x}_u \vee \bar{x}_{v_1} \vee \ldots \bar{x}_{v_d} \vee \bar{x}_{v_{d+1}})$ in $(d + 2)$-CNF formula which means all $u, v_1, \ldots, v_{d+1}$ cannot be colored all black or colored all white.

If the variable $x_u = 1$ means that the vertex u is colored black and the variable $x_u = 0$ means that the vertex u is colored white. It is not hard to see that F has a 0/1 satisfying assignment such that the number of ones is at most k if and only if the input graph G of the BD-dSB problem admits a vertex bipartition (B, W) with $V = B \uplus W$ such that both B and W are bounded-degree-d sets and $|B| \leq k$. Thus, to solve the BD-dSB problem can be reduced to solve the Minimum One $(d + 2)$-SAT problem.

The Minimum one $(d + 2)$-SAT problem can be solved in time $O^*((d + 2)^k)$ by the following algorithm where k is the number of true variables.

- For each clause $(x_1 \vee \cdots \vee x_{d+2})$, the algorithm branches $d+2$ cases, *i.e.,* for $i = 1, \ldots, d+2$, let $x_i = 1$ and $k := k - 1$.

The above algorithm for Minimum Ones $(d + 2)$-SAT runs in time $O^*((d + 2)^k)$ which shows that the s-plex bipartition problem considered in this paper can be solved in time $O^*((s + 1)^k)$, *i.e.,* fixed-parameter tractable for constant s. For $s = 2$, this algorithm runs in time $O^*(3^k)$.

In this paper, we design more customized fixed-parameter algorithm for 2-Plex Bipartition and improve the running time to be $O^*(2.4143^k)$ where $k = \min\{|V_1|, |V_2|\}$, (V_1, V_2) is a bipartition of V satisfying that both V_1 and V_2 are 2-plexes. By applying this algorithm as a subroutine, we give the first exact algorithm to find a Defective $(2, 1)$-Coloring with one of the two colors of minimum cardinality for a given graph with running time $O^*(1.5539^n)$ where n is the number of vertices in the input graph. In the following, we define a problem related to Bounded-Degree-1 Set Bipartition.

Color Constrained Bounded-Degree-1 Set Bipartitioning (CCBD-1SB)
Instance: A graph $G = (B \uplus U, E)$
Parameter: An integer $k \geq 0$
Question: Does there exist a vertex bipartition $U = (B', W')$ such that both $B \cup B'$ and W' are bounded-degree-1 sets and $|B \cup B'| \leq k$?

Notice that if $B = \emptyset$, the CCBD-1SB problem is equivalent to the BD-1SB problem. In the rest of the paper, we solve the CCBD-1SB problem.

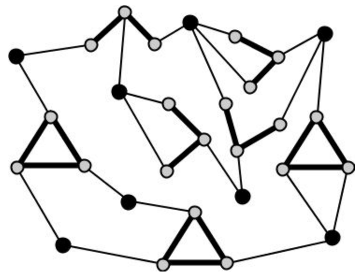

Figure 1 A partially black triple graph where grey nodes denote vertices in U and black nodes denote vertces in B. We use thick lines to denote edges with both endvertices in U and thin lines to denote edges with one endvertex in B and the other endvertex in U.

We close the section with some notation definitions. Let $G = (V, E)$ be a simple graph. For a vertex v in G, we use $N_G(v)$ to be the set of vertices adjacent to v. A path P_h in G is a path consisting of h vertices. We use C_h to denote a cycle of h vertices. If we use $(v_1, v_2, \ldots, v_h)$ to denote a P_h, then it means that $v_iv_{i+1} \in E$ for $1 \le i \le h-1$. If we use $(v_1, v_2, \ldots, v_h)$ to denote a C_h, then it means that $v_iv_{i+1} \in E$ for $1 \le i \le h-1$ and $v_1v_h \in E$. For a vertex set $X \subseteq V$, let $G[X] = (X, E(X))$ where $E(X) = \{uv \in E \mid u, v \in X\}$. For $X, Y \subseteq V$, use $X \uplus Y$ to be $X \cup Y$ satisfying $X \cap Y = \emptyset$.

2 A fixed-parameter algorithm for partially black triple graphs

In this section, we define a graph class called partial black triple graphs $\mathcal{C}$ and give a fixed-parameter algorithm for the Color Constrained Bounded-Degree-1 Set Bipartition problem in partial black triple graphs running in time $O(2^k)$.

▶ **Definition 1.** A graph $G = (V = B \uplus U, E)$ is called a partially black triple graph if the following conditions hold.

(i) Vertices in B are colored black and form an independent dominating set[2] in G.
(ii) Vertices in U are uncolored. Each vertex $v \in U$ is adjacent to exactly one black vertex in B.
(iii) Each connected component in $G[U]$ is either a P_3 or a C_3.

Next we define the Color Constrained Bounded-Degree-1 Set Bipartition problem in partially black triple graphs.

CCBD-1SB in Partially Black Triple Graphs
Instance: A partially black triple graph $G = (B \uplus U, E)$
Parameter: An integer $k \ge 0$
Question: Does there exist a vertex bipartition $U = B' \uplus W'$ such that both $B \cup B'$ and W' are both bounded-degree-1 sets and $|B \cup B'| \le k$?

Suppose that we use two colors black and white to color vertices U. A black-and-white coloring of U is said to be *feasible* if all black vertices in G form a bounded-degree-1 set and all white vertices in G form a bounded-degree-1 set.

[2] A vertex set $D \subseteq V$ is an *independent dominating set* in a graph $G = (V, E)$ if no two vertices in D are adjacent in G and each vertex $v \in V \setminus D$ is adjacent at least one vertex of D.

▶ **Lemma 2.** *Let $G = (B \uplus U, E)$ be a partially black triple graph that admits a vertex bipartition of $U = B' \uplus W'$ such that both $B \cup B'$ and W' are both bounded-degree-1 sets and $|B \cup B'| \leq k$. Then there exists $S \subseteq U$ of minimum cardinality such that $U \setminus S$ is a bounded-degree-1 set and $|S| \leq k - |B|$. Moreover, S can be found in polynomial time.*

Proof. Since G admits a bounded-degree-1 bipartition $(B \cup B', W')$ such that $|B'| \leq k - |B|$ and S is a subset of U of minimum cardinality such that $U \setminus S$ is a bounded-degree-1 set, we see that $|S| \leq |B'| \leq k - |B|$. According to the definition of partially black triple graph, each connected components of $G[U]$ is either a P_3 or a C_3. The set S can be obtained by picking exactly one vertex of degree one in each P_3 and exactly one vertex in each C_3 in $G[U]$. Thus the set S can found in polynomial time. This completes the proof. ◀

▶ **Lemma 3.** *Let $G = (B \uplus U, E)$ be a partially black triple graph. To determine whether there exists a vertex bipartition of $U = B' \uplus W'$ such that both $B \cup B'$ and W' are bounded-degree-1 sets and $|B \cup B'| \leq k$ can be done in time $O^*(2^{k-|B|})$.*

Proof. Let $S \subseteq U$ of minimum cardinality such that $G[U \setminus S]$ is of maximum degree one. By Lemma 2, if G admits a vertex bipartition of $U = B' \uplus W'$ such that both $B \cup B'$ and W' are both bounded-degree-1 sets and $|B \cup B'| \leq k$, then $|S| \leq k - |B|$. Thus, if $|S| > k - |B|$, our algorithm can simply return that no such bipartition of U exists in polynomial time.

Suppose that $|S| \leq k - |B|$. Our algorithm enumerates $2^{|S|} \leq 2^{k-|B|}$ possibilities to partition $S = S_B \uplus S_W$ where $S_B \subseteq B'$ and $S_W \subseteq W'$. Notice that S is an independent set in G. This implies that S_B and S_W are both independent sets in G. If $S_B \cup B$ is not a bounded-degree-1 set, then the partition of S is not feasible. Assume that $S_B \cup B$ is a bounded-degre-1 set in G. We say vertices in $B \cup S_B$ are colored black and vertices in S_W are colored white.

Notice that vertices in S are collected by picking exactly one vertex of degree one in each P_3 and exactly one vertex in each C_3 in $G[U]$. Let $U' = U \setminus S$. It is easy to see that $G[U']$ is a 1-regular graph.

In the rest of the proof, we call vertices in U' uncolored and we call U' the uncolored set. Once an uncolored vertex is colored black or white, it is removed from the uncolored set U'. The rest of the problem is to color vertices U' black or white.

If the following cases exist, we can simply color a vertex white or black.

- If $v \in U'$ is adjacent to two black vertices, then v must be colored white. Remove v from U'.
- If $v \in U'$ is adjacent to two white vertices, then v must be colored black. Remove v from U'.
- If $v \in U'$ is adjacent to a black vertex x and x has a black neighbor, then v must be colored white. Remove v from U'.
- If $v \in U'$ is adjacent to a white vertex x and x has a white neighbor, then v must be colored black. Remove v from U'.
- If there exists $v \in U'$ of degree two in G, $N_G(v) = \{x, y\}$, satisfying that x is black, y is white, and vertices in $N_G(y) \setminus \{v\}$ are all black, then color v white. Remove v from U'.

Suppose the above cases do not exist. We see that $G[U']$ is a 1-regular graph and every vertex $v \in U'$ is either of degree two or three in G. Moreover, in G each uncolored vertex $v \in U'$ of degree two is adjacent to an uncolored vertex and a black vertex and each uncolored vertex $v \in U'$ of degree three is adjacent to an uncolored vertex, a black vertex, and a white vertex.

▶ **Claim 4.** *If each uncolored vertex $v \in U'$ of degree two is adjacent to an uncolored vertex and a black vertex; and each uncolored vertex $v \in U'$ of degree three is adjacent to an uncolored vertex, a black vertex, and a white vertex, then any feasible coloring of $U' = B' \uplus W'$ must satisfy $|B'| = |W'|$.*

Proof of Claim 4. Let $u, v \in U'$ be two adjacent vertices in G. Note that both u, v are adjacent to exactly one black vertex and at most one white vertex. If both u and v are colored black, then there exists a P_3 (u, v, x) such that all three vertices are colored black where x is the white vertex adjacent to u, a contradiction to the fact that this is a feasible coloring. Since at least one of u and v is adjacent to a white vertex, say u, if both u and v are colored white, then there exists a P_3 (u, v, x) such that all three vertices are colored white where x is the white vertex adjacent to u, a contradiction to the fact that this is a feasible coloring. Thus, one of u and v must be colored black and the other must be colored white. This shows that any feasible coloring of $U' = B' \uplus W'$ must satisfy $|B'| = |W'|$. This completes the proof. ◀

Because any two adjacent vertices $u, v \in U'$ must be colored different colors and any feasible coloring of $U' = B' \uplus W'$ must satisfy $|B'| = |W'|$, we can formulate the rest problem as a 2-SAT problem. Here is the way to formulate the problem with a 2-CNF formula.

- For each $v \in U'$, create a variable x_v.
- For any two adjacent vertices $u, v \in U'$, create two clauses $(x_u \vee x_v)$ and $(\bar{x}_u \vee \bar{x}_v)$ in the 2-CNF formula. The two clauses mean that x_u and x_v cannot be both true or both false. We use $x_u = 1$ to denote that u is colored black and $x_u = 0$ to denote that u is colored white.
- For any black vertex z, if there exists $u, v \in U'$ and (u, v, z) is a P_3 in G, create a clause $(\bar{x}_u \vee \bar{x}_v)$ in the 2-CNF formula. This means x_u and x_v cannot be both true. Both u and v cannot be both colored black.
- For any white vertex z, if there exists $u, v \in U'$ and (u, v, z) is a P_3 in G, create a clause $(x_u \vee x_v)$ in the 2-CNF formula. This means x_u and x_v cannot be both false. Both u and v cannot be both colored white.

It is not hard to see that the 2-SAT formula returns true if and only if U' admits a feasible coloring with $U' = B' \uplus W'$ satisfying $|B'| = |W'|$. Since 2-SAT can be solved in polynomial time [22], to determine whether U admits a feasible 2-coloring can be done in polynomial time.

Notice that the algorithm enumerates all $2^{|S|} \leq 2^{k-|B|}$ possibilities of black and white colorings of vertices in S. This produces at most $2^{k-|B|}$ subproblems. From the above cases analysis, all these coloring subproblems can be solved in polynomial time. This shows that to determine whether there exists a vertex bipartition of $U = B' \uplus W'$ such that both $B \cup B'$ and W' are bounded-degree-1 sets and $|B \cup B'| \leq k$ can be done in time $O^*(2^{k-|B|})$. This completes the proof. ◀

3 A fixed-parameter algorithm for general graphs

In this section, we give a branch-and-reduce algorithm for the CCBD-1SB problem to solve the CCBD-1SB problem running in time $O^*(2.4143^k)$ where k is the number of black vertices. Given an input graph $G = (B \uplus U, E)$, of the CCBD-1SB problem, the algorithm outputs a vertex bipartition $U = (B', W')$ such that both $B \cup B'$ and W' are bounded-degree-1 sets, if the bipartition exists. Note that U is called the set of uncolored vertices, vertices in U are called *uncolored*, and vertices in B are called black. The algorithm consists of reduction rules and branching rules and repeats the execution of the first applicable rule in the sequence. Thus, inside a given case, the hypotheses of all previous rules are assumed to be inapplicable. Notice that if we say an uncolored vertex is colored black or white in the algorithm, then the vertex will be removed from the uncolored set U but the vertex remains in the graph G. We say a vertex is removed from G that means it is deleted from the input graph.

3.1 Reduction Rules

Let $G = (B \uplus U, E)$ be the input graph of the CCBD-1SB problem and k be the input parameter. We first give some reduction rules applied in the fixed-parameter algorithm.

Reduction Rules

- **Too many colored neighbor rule.** If there exists an uncolored vertex having two black and two white neighbors, then return "No."
- **Same color P_3 rule.** If there exists a P_3 with three vertices getting the same color, then return "No."
- **Two adjacent vertex rule.** If black (white) vertices u, v are adjacent, then any vertex x adjacent to u, v must be colored white (black). Remove x from U and remove u, v from G if both u, v are white.
- **Isolated white rule.** If a white vertex v only adjacent to black vertices, then remove v from G.
- **Two same color neighbors rule.** If there is an uncolored vertex $v \in U$ with two black (white) neighbors, then v must be colored white (black). Remove v from U.
- **One uncolored rule.** If there is an uncolored vertex $v \in U$ without uncolored neighbors satisfying one of the following conditions,
  1. v has no white neighbors, or
  2. v has exactly one white neighbor u and at most one black neighbor, and all neighbors of u are colored black.

  Then v must be colored white. Remove v from U.
- **Two uncolored rule.** If there are two adjacent uncolored vertices $u, v \in U$ in G having no white neighbors and no other uncolored neighbors, then u, v must be colored white. Remove u, v from U. Notice that if G has no white vertices and the **Two uncolored rule** is not applicable, there is no connected component which is a P_2 in $G[U]$.
- **No black neighbor triple rule.** If there exists three uncolored vertices $x, y, z \in U$ inducing a P_3 or a C_3 satisfying that x is not adjacent to any black vertex and x, y, z has no uncolored neighbors other than x, y, z and x, y, z are not adjacent to any white vertex in G, then color x black and color y, z white. When the **No black neighbor triple rule** can not be applied, we see that each vertex in a connected component with exactly three vertices in $G[U]$ has a black neighbor in G.

▶ **Lemma 5.** *Two adjacent vertex rule is valid.*

Proof. Suppose that both u and v are black. If a black vertex x is adjacent to u or v, then there exists a P_3 (u, v, x) or (v, u, x) such that all three vertices are colored black, a contradiction to the fact that this is a feasible coloring. Thus, all vertices adjacent to u, v must be colored white. Suppose that both u and v are white. If a white vertex x is adjacent to u or v, then there exists a P_3 (u, v, x) or (v, u, x) such that all three vertices are colored white, a contradiction to the fact that this is a feasible coloring. Thus, all vertices adjacent to u, v must be colored black. This completes the proof. ◀

▶ **Lemma 6.** *Isolated white rule is valid.*

Proof. Let v be a white vertex only adjacent to black vertices and all its neighbors are colored. Since the rest of uncolored vertices to be colored black or white are not affected by the color of v, v can be removed from G safely. This completes the proof. ◀

▶ **Lemma 7.** *Two same color neighbors rule is valid.*

Proof. Suppose that v is an uncolored vertex having two black neighbors. If v is black, then the set of black vertices is not a bounded-degree-1 set because the degree of v is two in the induced subgraph of black vertices. Suppose that v is an uncolored vertex having two white neighbors. If v is white, then the set of white vertices is not a bounded-degree-1 set because the degree of v is two in the induced subgraph of white vertices. This completes the proof. ◀

▶ **Lemma 8.** *One uncolored rule is valid.*

Proof. Let v be an uncolored vertex having no uncolored neighbors, *i.e.,* v is an isolated vertex in $G[U]$. Notice that v is adjacent to at most one black vertex in G. Let $\mathcal{P}$ be an optimal solution of CCBD-1SB with minimum number of black vertices and the number of black vertices is at most k. Suppose that v is not adjacent to any white vertex and v is colored black in $\mathcal{P}$. We then obtain a solution $\mathcal{P}'$ from $\mathcal{P}$ by recoloring v white, a contradiction to the assumption that $\mathcal{P}$ is an optimal solution with minimum number of black vertices. Suppose that v is adjacent to exactly one white neighbor u and u is only adjacent to black vertices. If v is colored black in $\mathcal{P}$, we then obtain a solution $\mathcal{P}'$ from $\mathcal{P}$ by recoloring v white, a contradiction to the assumption that $\mathcal{P}$ is an optimal solution with minimum number of black vertices. This completes the proof. ◀

▶ **Lemma 9.** *Two uncolored rule is valid.*

Proof. Let x, y be two adjacent uncolored vertices having no white neighbors and no other uncolored neighbors, *i.e.,* $\{u, v\}$ induces a P_2 in $G[U]$ and forms a connected component in $G[U]$. Note that u and v are only adjacent to black vertices. Let $\mathcal{P}$ be an optimal solution of CCBD-1SB with minimum number of black vertices and the number of black vertices is at most k. If one of u, v is black, say u, a solution $\mathcal{P}'$ can be obtained from $\mathcal{P}$ by recoloring u white, a contradiction to the assumption that $\mathcal{P}$ is an optimal solution with minimum number of black vertices. This completes the proof. ◀

▶ **Lemma 10.** *No black neighbor triple rule is valid.*

Proof. Let $\{x, y, z\}$ be a connected component in $G[U]$ satisfying that x is not adjacent to any black vertex in G. Notice that $\{x, y, z\}$ induces either a P_3 or a C_3 in $G[U]$ and y and z are adjacent to at most one black neighbor and no white vertex is adjacent to x, y, z in G. Suppose that there is a solution $\mathcal{P}$ of CCBD-1SB that x is white. Since $\{x, y, z\}$ induce a P_3 or a C_3, at least one of y and z must be black in $\mathcal{P}$, otherwise there is a degree-two vertex in the induced subgraph of white vertices. We see that a feasible solution $\mathcal{P}'$ can be obtained from $\mathcal{P}$ by recoloring x black and y and z white. Moreover, $\mathcal{P}'$ and $\mathcal{P}$ have same number of black vertices. This completes the proof. ◀

3.2 Branching Rules

Suppose all the reduction rules are not applicable. The algorithm then applies the following branching rules.

- **One white neighbor rule.** There is an uncolored vertex v having exactly one white neighbor u and satisfying that v or u has an uncolored neighbor x. Then the algorithm branches on each of the following cases:
  - v is black and $k := k - 1$; or
  - v is white and x is black and $k := k - 1$.

  The branching vector of this rule is $(1, 1)$ and the branching number is 2. Notice that if **One white neighbor rule** is not applicable, no white vertices is adjacent to an uncolored vertex. Moreover, if the **One uncolored rule** and **One white neighbor rule** can not be applied, there is no isolated vertices in $G[U]$.
- **Standard branching rule.** There is an uncolored vertex v with at least three uncolored neighbors $v_1, v_2, \ldots, v_h$. Then the algorithm branches on each of the following cases:
  - v is black and $k := k - 1$;
  - v is white and $v_1, v_2, \ldots, v_h$ are black and $k := k - h$; or
  - branch on each $1 \leq i \leq h$, let v, v_i be white and $\{v_1, v_2, \ldots, v_h\} \setminus \{v_i\}$ be black and $k := k - h + 1$.

  The branching vector of this rule is $(1, h, h-1, \ldots, h-1)$ where the cardinality of the branching vector is $h + 2$ and $h \geq 3$. The worst cases happens when $h = 3$ and its branching vector is $(1, 3, 2, 2, 2)$ and the branching number is 2.4143. Note that if the **Standard branching rule** is not applicable, the maximum degree of $G[U]$ is at most two.
- **Four path end rule.** There exists a P_4 (v_1, v_2, v_3, v_4) in $G[U]$ satisfying that v_1 has only one uncolored vertex v_2 and v_2 has only two uncolored neighbors v_1, v_3. The algorithm branches on each of the following cases:
  - v_2 is black and $k := k - 1$;
  - v_1, v_2 are white and v_3 is black and $k := k - 1$; or
  - v_2, v_3 are white and v_1, v_4 are black and $k := k - 2$.

  The worst case branching vector of this rule is $(1, 1, 2)$ and the branching number is 2.4143. When all the reduction rules, **One white neighbor rule**, **Standard branching rule**, and **Four path end rule** are not applicable, we see that any path in $G[U]$ has exactly three vertices.
- **At least four cycle rule.** There exists a C_h $(v_1, v_2, v_3, \ldots, v_h)$ in $G[U]$ where $h \geq 4$. Then the algorithm branches on each of the following cases:
  - v_2 is black and $k := k - 1$;
  - v_2 is white and v_1, v_3 are black and $k := k - 2$;
  - v_2, v_3 are white and v_1, v_4 are black and $k := k - 2$; or
  - v_1, v_2 are white and v_3, v_h are black and $k := k - 2$.

  The worst case branching vector of this rule is $(1, 2, 2, 2)$ and the branching number is 2.3028. When the **At least four cycle rule** can not be applied, we see that any cycle in $G[U]$ has at most three vertices.

  When all reduction rules and all the above branching rules are not applicable, we see that the maximum degree of $G[U]$ is two. Moreover, all connected components in $G[U]$ are of three vertices, *i.e.,* they are either P_3 or C_3. Since the reduction rule, **No black neighbor triple rule** is not applicable, each of the uncolored vertex is adjacent to exactly one black vertex. By definition, the remaining graph is a partially black triple graph.

- **Partially black triple rule.** If $G = (B \uplus U, E)$ is a partially black triple graph, use the algorithm in Section 2 to solve the problem in time $O^*(2^{k-|B|})$.

Now we prove the correctness of the branching rules.

▶ **Lemma 11.** *One white neighbor rule is valid.*

Proof. Let v be an uncolored vertex having exactly one white neighbor u. Let x be an uncolored vertex adjacent to v or u. We see that v is either colored black or colored white in any optimal solution. If v is colored white, then x cannot be colored white, otherwise the set of white vertices in the optimal solution is not a bounded-degree-1 set, a contradiction. Thus, if v is colored white, then x must be colored black. This completes the proof. ◀

▶ **Lemma 12.** *Standard branching rule is valid.*

Proof. For an uncolored vertex v having h uncolored neighbors, $h \geq 3$, the Standard branching rule branches all $(h+2)$ possibilities to color v and all its uncolored neighbors. In an optimal solution, either v is colored black or colored white. If v is white, either all its neighbors are all colored black or exactly one of its neighbor is colored white. By Lemma 5, if v and one of its neighbor are colored white, then all the uncolored neighbors of v must be colored black. This completes the proof. ◀

▶ **Lemma 13.** *Four path end rule is valid.*

Proof. Notice that if all the reduction rules and **One white neighbor rule** and **Standard branching rule** are not applicable, then G has no white vertices, each uncolored vertex v in G has at most one black neighbor, and each vertex in $G[U]$ is of degree at most two. Suppose that (v_1, v_2, v_3, v_4) is a P_4 in G and all of them are uncolored. Let $\mathcal{P}$ be an optimal solution of CCBD-1SB with minimum number of black vertices at most k. We see that v_2 is either colored black or white. Suppose that v_2 is colored white in the optimal solution $\mathcal{P}$. If v_1 is colored white in the optimal solution $\mathcal{P}$, then v_3 must be colored black. If v_3 is colored white in the optimal solution $\mathcal{P}$, then both v_1 and v_4 must be colored black. Suppose that v_2 is colored white and both v_1 and v_3 are colored black in the optimal solution $\mathcal{P}$. Since all the neighbors of v_2 are colored black and v_2 has no white neighbors, we can always obtain a solution $\mathcal{P}'$ from $\mathcal{P}$ by recoloring v_1 white. We see that $\mathcal{P}'$ has less black vertices than $\mathcal{P}$, a contradiction to the assumption that $\mathcal{P}$ is an optimal solution with minimum number of black vertices. This completes the proof. ◀

▶ **Lemma 14.** *At least four cycle rule is valid.*

Proof. Notice that if all the reduction rules and **One white neighbor rule**, **Standard branching rule**, and **Four path end rule** are not applicable, then G has no white vertices, each uncolored vertex v in G has at most one black neighbor, each vertex in $G[U]$ is of degree at most two, and $G[U]$ consists of cycles and P_3s. Suppose that $\{v_1, v_2, v_3, \ldots, v_h\}$ induces a cycle in G such that $(v_i, v_{i+1}), (v_1, v_h) \in E$ for $1 \leq i \leq h-1$, and all of them are uncolored. Let $\mathcal{P}$ be an optimal solution of CCBD-1SB with minimum number of black vertices at most k. We see that v_2 is either colored black or white. Suppose that v_2 is colored white in $\mathcal{P}$. Then either both v_1 and v_3 are black in $\mathcal{P}$ or one of v_1 and v_3 is colored white in $\mathcal{P}$. If v_1 is colored white in the optimal solution $\mathcal{P}$, then v_h and v_3 must be colored black. If v_3 is colored white in the optimal solution $\mathcal{P}$, then v_1 and v_4 must be colored black. This completes the proof. ◀

Notice the worst branching number 2.4143 is obtained from **Standard branching rule** and **Four path end rule**. We now conclude this section with the following theorem and corollaries.

▶ **Theorem 15.** *The CCBD-1SB problem can be solved in $O^*(2.4143^k)$ time.*

▶ **Corollary 16.** *The* 2-PLEX BIPARTITION *problem and the* BOUNDED-DEGREE-1 SET BIPARTITION *problem can be solved in $O^*(2.4143^k)$ time.*

▶ **Corollary 17.** *The* 2-PLEX BIPARTITION *problem and the* BOUNDED-DEGREE-1 SET BIPARTITION *problem can be solved in $O^*(1.5539^n)$ time where n is the number of vertices in the input graph.*

Proof. By Corollary 16, the 2-PLEX BIPARTITION problem and the BOUNDED-DEGREE-1 SET BIPARTITION problem can be solved in $O^*(2.4143^k)$ time. According to the fact that $k \leq n/2$, we can design an exact algorithm by using the fixed-parameter algorithm for the 2-PLEX BIPARTITION problem and the BOUNDED-DEGREE-1 SET BIPARTITION problem as a subroutine. The running time of the exact algorithm is $O^*(2.4143^{n/2}) = O^*(1.5539^n)$. This completes the proof. ◀

4 Concluding remarks

In this paper, we give a fixed-parameter algorithm to solve the 2-plex bipartition problem in time $O^*(2.4143^k)$ where $k \leq n/2$ is an input parameter. It is of interesting to see whether there exist more efficient fixed-parameter algorithms to solve the s-plex bipartition problem for a constant $s \geq 2$. Moreover, it is even more interesting to see whether there exist fixed-parameter algorithms to solve the s-plex t-partition problem that asks whether the vertices of input the graph can be partitioned into t parts such that each part is an s-plex.

References

1. J. Andrews and M. Jacobson, On a generalization of chromatic number, *Congressus Numerantium*, **47** (1985), 33–48.
2. B. Balasundaram, Cohesive subgroup model for graph-based text mining, *Proceedings of the 2008 IEEE Conference on Automation Science and Engineering*, pp. 989–994, 2008.
3. B. Balasundaram, S. Butenko, and I. V. Hicks, Clique relaxations in social network analysis: The maximum k-plex problem, *Operations Research* 59 (2011), pp. 133–142.
4. R. van Bevern, H. Moser, and R. Niedermeier, Approximation and tidying–a problem kernel for s-plex cluster vertex deletion, *Algorithmica* 62 (2012), pp. 930–950.
5. M.-S. Chang, L.-J. Hung, and P.-C. Su, Exact and fixed-parameter algorithms for problems related to 2-plex, *Proceedings of the 2011 International Computer Science and Engineering Conference (ICSEC 2011)*, pp. 203–208, 2011.
6. M.-S. Chang, L.-H. Chen, L.-J. Hung, Y.-Z. Liu, P. Rossmanith, and S. Sikdar, An $O^*(1.4658^n)$-time exact algorithm for the maximum bounded-degree-1 set problem, *Proceedings of the 31st Workshop on Combinatorial Mathematics and Computation Theory*, pp. 9–18, 2014.
7. M.-S. Chang, L.-H. Chen, L.-J. Hung, P. Rossmanith, and P.-C. Su, Fixed-parameter algorithms for vertex cover P_3, *Discrete Optimization*, 19 (2016), 12–22.
8. L. Cowen, R. Cowen, and D. Woodall, Defective colorings of graphs in surfaces: partitions into subgraphs of bounded valence, *Journal of Graph Theory*, 10 (1986), 187–195.

9 R. Cowen, Some connections between set theory and computer science, in G. Gottlob, A. Leitsch, and D. Mundici (eds.), *Proceedings of the third Kurt Godel colloquium on Computational Logic and proof theory*, Lecture Notes in Computer Science, pp. 14–22, 1993.

10 L. Cowen, W. Goddard, and C. E. Jesurum, Defective coloring revisited, *Journal of Graph Theory*, 24 (1997), pp. 205–219.

11 M. Cygan, F.V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, *Parameterized algorithms*, Springer, 2015.

12 R. G. Downey and M. R. Fellows, *Fundamentals of Parameterized Complexity*, Springer, 2013.

13 F. Fomin and D. Kratsch, *Exact Exponential Algorithms*, Springer, 2010.

14 M. Frick, A survey of (m, k)-colorings, in J. Gimbel, J. W. Kennedy, and L. V. Quintas (eds.), *Quo Vadis, Graph Theory?*, volume 55 of Annals of Discrete Mathematics, pp. 45–58. Elsevier Science Publisher, New York, 1993.

15 W. Goddard, Acyclic colorings of planar graphs, *Discrete Mathematics*, 91 (1991), 91–94.

16 F. Harary and K. Jones, Conditional colorability II: Bipartite variations, *Congressus Numerantium*, 50 (1985), 205–218.

17 T. Jensen and B. Toft, *Graph Coloring Problems*, John Wiley and Sons, New York (1995).

18 F. T. Leighton, A graph coloring algorithm for large scheduling problems, *Journal of Research of the Notional Bureau of Standards*, Vol. 84, No.6, November- December, 1979.

19 J. M. Lewis and M. Yannakakis, The node-deletion problem for hereditary properties is NP-complete, *Journal of Computer and System Sciences*, 20 (1980), 219–230.

20 V. Lotfi, A graph coloring algorithm for large scale scheduling problems, *Computers & Operations Research* 13 (1986), 27–32.

21 L. Lovász, On decomposition of graphs, *Studia Scientiarum Mathematicarum Hungarica* 1 (1966), pp. 237–238.

22 C. Papadimitriou, On selecting a satisfying truth assignment, In *Proceedings of the 32nd Annual IEEE Symposium on Foundations of Computing (FOCS 1991)*, pp. 163–169, 1991.

23 K. Poh, On the linear vertex-arboricity of a planar graph, *Journal of Graph Theory*, 14 (1990), 73–75.

24 S. B. Seidman and B. L. Foster, A graph-theoretic generalization of the clique concept, *The Journal of Mathematical Sociology* 6 (1978), pp. 139–154.

25 D. Woodall, Improper colourings of graphs, in R. Nelson and R. J. Wilson (eds.), *Graph Colourings*, Longman Scientific and Technical (1990).