Received: 19 August 2016 | Accepted: 20 April 2018
DOI: 10.1002/net.21826

RESEARCH ARTICLE

WILEY

Integer programming formulations for minimum deficiency interval coloring

Merve Bodur[1] | **James R. Luedtke**[2]

[1]Department of Mechanical and Industrial Engineering, University of Toronto, Toronto

[2]Department of Industrial and Systems Engineering, University of Wisconsin-Madison, Madison

Correspondence
Merve Bodur, Department of Mechanical and Industrial Engineering, University of Toronto, Toronto.
Email: bodur@mie.utoronto.ca

Funding information
National Science Foundation, Grant/Award Number: SES-1422768; Office of Naval Research, Grant/Award Number: N00014-15-1-2268

Abstract

A proper edge-coloring of a given undirected graph with natural numbers identified with colors is an *interval (or consecutive) coloring* if the colors of edges incident to each vertex form an interval of consecutive integers. Not all graphs admit such an edge-coloring and the problem of deciding whether a graph is interval colorable is NP-complete. For a graph that is not interval colorable, determining a graph invariant called the (minimum) *deficiency* is a widely used approach. Deficiency is a measure of how close the graph is to have an interval coloring. The majority of the studies in the literature either derive bounds on the deficiency of general graphs or calculate the deficiency of graphs belonging to some special graph classes. In this work, we derive integer programming formulations of the *Minimum Deficiency Problem* which seeks to find the exact deficiency value of a graph, given a bound on the number of colors that can be used. We further enhance the formulation by introducing a family of valid inequalities. Then, we solve our model via a *branch-and-cut algorithm*. Our computational study on a large set of random graphs illustrates the strength of our formulation and the efficiency of the proposed approach.

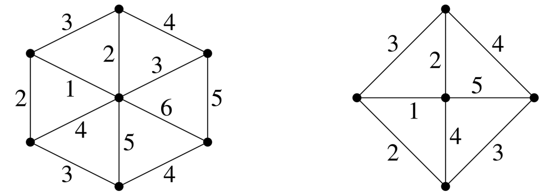
KEYWORDS

column generation, cutting planes, integer programming, interval (consecutive) edge-coloring, minimum deficiency problem

1 | INTRODUCTION

Coloring problems are an extensively studied class of problems in graph theory. They arise in many applications such as scheduling and frequency allocation [30]. Besides their practical importance, they have many theoretical implications as they are useful in defining graph subclasses and provide graph invariants such as vertex chromatic number and edge chromatic number.

A proper edge-coloring in a graph is an assignment of colors to edges such that no two adjacent edges have the same color. When colors are labeled with the natural numbers, an *interval coloring* is a proper edge-coloring in which for each vertex of the graph, the colors assigned to edges incident to the vertex form an interval of consecutive integers. An interval coloring that uses K colors is referred to as an interval K-coloring. If there exists an interval coloring for a graph (with K colors), the graph is called *interval colorable* (interval K-colorable). For example, an interval coloring of the wheel with seven nodes (W_7) is provided in Figure 1(a) where the color numbers are given next to the edges. However, not all graphs are interval colorable, such as W_3 and W_5 (e.g., see W_5 in Figure 1(b)) [18]. For those graphs that are not interval colorable, a natural approach is to define a graph invariant to measure how close the graph is to admit an interval coloring. This invariant is called the (minimum) *deficiency* of a



(a) An interval 6-coloring of W_7. (b) A 5-coloring of W_5 with deficiency 1.

FIGURE 1 Interval coloring and deficiency examples

graph and thus the problem of finding that invariant is called the *Minimum Deficiency Problem* (MinDef). The goal in (MinDef) is to identify a proper edge-coloring that misses the least number of colors in the "intervals" of colors assigned to each vertex (i.e., the sets of colors assigned to incident edges of each vertex). A coloring of W_5 having the (minimum) deficiency of one is provided in Figure 1(b). Note that the "interval" of the middle vertex, $\{1, 2, 4, 5\}$, is missing one color, thus yielding the deficiency of one. Formal definitions of deficiency are provided in Section 1.1.

There are several extensions of interval coloring. For *cyclically interval coloring*, a proper edge-coloring using up to a given number of colors with the following property is desired: For each vertex, either the set of colors used on edges incident to the vertex or the set of colors not used on edges incident to the vertex forms an interval of integers [26, 28]. *One-sided interval coloring* is designed for bipartite graphs with the aim of finding a proper edge-coloring where the "interval property" is required to be satisfied only for the vertices in one partite vertex set [27]. In some problems, the interval property is replaced with "no more than one gap property" [39]. We do not explore these extensions in this paper, although our approach can be easily extended to such problems.

Many classes of graphs are interval colorable (when the number of colors used is not restricted), and our numerical experiments suggest that nearly all randomly generated graphs are interval colorable. Thus, a natural extension is to find a (near) minimum number of colors required to obtain an interval coloring, that is, to find a small or minimum value of K such that the graph is interval K-colorable. We present integer programming based methods for finding a minimum deficiency interval K-coloring in a graph. Thus, by using a large enough value for K, this provides a formulation of (MinDef). On the other hand, the formulation can also be used to find the smallest value of K for which an interval coloring exists (if any), by starting with a small value of K, and increasing K until an interval coloring is found.

The interval coloring problem has potential applications in task scheduling, especially in constructing timetables with "compactness requirements" [4, 30]. In particular, it is useful for scheduling problems without waiting or idle times. For instance, a given set of job interviews between some firms and candidates can be modeled as a bipartite graph, with one partite vertex set representing the firms, the other representing candidates, and edges corresponding to the interviews. In this particular case, the schedule (timetable) may be interpreted as an edge-coloring of the graph with natural numbers corresponding to assigned time slots. Then, an interval coloring of the graph provides a schedule where neither firms nor candidates wait between their meetings. If the idle times and waiting times are not prohibited but their number has to be minimized, the problem is then equivalent to (MinDef). That is, if we attach a set of pendant edges to the bipartite graph and obtain an interval coloring for this extended (bipartite) graph, then the colors assigned to the pendant edges attached to a firm node and a candidate node correspond to the idle time slots of the firm and the slots that the candidate has to wait, respectively. Therefore, a solution of (MinDef) problem yields a schedule with the minimum number of interruptions. Giaro and Kubale [15] consider this scheduling problem in the case of open, flow, and mixed shops. The first case is equivalent to interval coloring, while the other two require some additional restrictions on the interval coloring.

To the best of our knowledge, (MinDef) has been addressed from the integer programming (IP) point of view only by Altinakar et al. [1], who introduce three variants of a natural IP formulation of the problem. They empirically test the efficacy of these formulations, and compare them with a natural constraint programming (CP) model. They observe that CP outperforms the IP models. However, these methods can only solve very small instances (with less than 10 vertices) to optimality.

In this paper, we start with an IP formulation of (MinDef) provided by Altinakar et al. [1], which is an *edge-based* formulation (i.e., decision variables correspond to the edges). We prove that its linear programming (LP) relaxation bound is always zero. Then, we propose an alternative edge-based IP formulation with a stronger LP relaxation. As the *matching-based* formulations usually provide stronger relaxations than the edge-based formulations for the standard edge-coloring problem, we improve this model further by deriving valid inequalities using the same observation that edge-colorings correspond to a covering of the edges by matchings. We suggest a *branch-and-cut algorithm* to solve the resulting strengthened formulation. We also provide

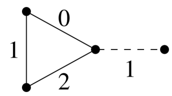


FIGURE 2 An interval coloring of K_3 augmented by one pendant edge

a matching-based formulation of (MinDef) which can be solved via a *column generation* algorithm. We conduct a numerical study on a large set of randomly generated graphs to compare the performance of our IP formulations, together with the CP formulation given by Altinakar et al. [1]. We find that our branch-and-cut algorithm performs significantly better than the other approaches (especially for dense graphs which usually have high deficiencies), finding good feasible solutions and providing very strong relaxation bounds, thus proving the optimality of the solutions significantly faster for a fixed small value of K. In addition, we demonstrate that this approach is effective at identifying small values of K for which the graph is interval K-colorable.

We next review some relevant definitions in Section 1.1 and related literature in Section 1.2.

1.1 | Preliminaries

Let $G = (V, E)$ be an undirected graph with vertex set V and edge set E. We assume that G is finite, simple, and connected. We denote the number of vertices in G, the maximum degree of G, and the chromatic index of G by n, Δ, and χ', respectively. For a vertex $v \in V$, we use $d(v)$ and $N(v)$ to represent the degree of v and the neighborhood of v (i.e., the set of vertices adjacent to v), respectively. We start with two alternative definitions of the term "deficiency".

Definition 1. The *deficiency of graph G* is the minimum number of *pendant edges* whose attachment to G forms an interval colorable supergraph of G, where a pendant edge is an edge having one endpoint in G and the other endpoint as a new vertex of degree one in the supergraph.

Figure 2 presents a supergraph of the complete graph of order three (K_3) obtained by attaching one pendant edge and an interval edge-coloring of the supergraph. Solid edges belong to the original graph K_3, while the dashed edge is a pendant edge. The color numbers are given next to the edges. Note that K_3 is not interval colorable, but the supergraph obtained by augmenting K_3 with one pendant edge is interval colorable. Therefore, the deficiency of K_3 is one.

The second interpretation of the deficiency requires several more definitions:

Definition 2. Given a finite subset A of $\mathbb{N}$, the *deficiency of A*, denoted by $\text{def}(A)$, is the number of integers in the interval $[\min(A), \max(A)]$ not belonging to A, that is,

$$\text{def}(A) = \max(A) - \min(A) - |A| + 1,$$

where min and max operators return the smallest and the largest elements in the set, respectively.

Note that if $\text{def}(A) = 0$, A is an *interval*.

Definition 3. Let $c : E \to \mathbb{N}$ be a proper edge-coloring of G and let $v \in V$. The *deficiency of coloring c at vertex v*, denoted by $\text{def}(G, c, v)$, is the deficiency of the set of colors assigned to edges incident to v under c.

Definition 4. Let $c : E \to \mathbb{N}$ be a proper edge-coloring of G. The *deficiency of coloring c* is the sum of deficiencies of all vertices in G under c, that is,

$$\text{def}(G, c) = \sum_{v \in V} \text{def}(G, c, v).$$

Definition 5. The *deficiency of graph G* is the minimum of the deficiencies of all possible proper edge-colorings of G, that is,

$$\mathrm{def}(G) = \min_{c \in C} \mathrm{def}(G, c),$$

where C represents the set of all proper edge-colorings of G.

The equivalence of the two definitions of the deficiency, that is, Definition 1 and Definition 5, is given by Giaro et al. [18].

Lastly, we note that an edge-coloring c of G using K different colors is called a K-edge-coloring of G. Alternatively, the number of colors, K, is referred to as the *span* of c. If there exists a K-edge-coloring with deficiency zero, then G is called *interval K-colorable*. The *minimum (maximum) span* is defined as the minimum (maximum) of the spans among all interval colorings of G.

1.2 | Literature review

The edge-coloring problem is first studied by Tait [44] in 1880 to prove the well-known "Four-Color Conjecture". Vizing [45] proves that any simple graph can be colored with $\Delta + 1$ colors, while König [29] shows for bipartite graphs that Δ colors are sufficient. Nemhauser and Park [34] introduce the first IP formulation for the edge-coloring problem which is a "matching-based formulation" in that the decision variables of the formulation correspond to matchings in the graph. Later, Lee and Leung [31] present a "multi-matching formulation" for the edge-coloring problem.

The interval coloring problem was first introduced by Asratian and Kamalian [5], where they show that deciding whether a regular graph is interval colorable is NP-complete. Sevastjanov [43] proves that it is also NP-complete to decide if a given bipartite graph admits an interval coloring. Giaro and Cubale [13] strengthen this result by showing that the problem of deciding interval Δ-colorability of a bipartite graph is easy if $\Delta \leq 4$ and becomes NP-complete for $\Delta \geq 5$.

Some restricted graph classes have been shown to be interval colorable, and the minimum and maximum spans are known for some other graph classes. All trees, complete bipartite graphs [4, 25], regular bipartite graphs [5, 6], doubly convex bipartite graphs [4, 23], grids [14], fans graphs with $n > 3$ [18], Mobius ladders [36], n-dimensional cubes [37], 2-processor and 3-processor bipartite graphs [17], bipartite cacti [16], outerplanar bipartite graphs [15], $(2, \Delta)$-biregular bipartite graphs [22, 24] and some classes of (3,4)-biregular bipartite graphs [3, 40, 46] are interval colorable. Exact values of the minimum and maximum span parameters are proven for some classes of trees, complete bipartite graphs [5, 25] and Möbius ladders [36]. For general graphs, the bounds on these parameters are examined [1, 6, 18].

The deficiency of graphs is first studied by Giaro et al. [18]. Although it is NP-hard to determine the deficiency of a graph in general [13], exact values of the deficiency have been determined for some special families of graphs such as cycles, complete graphs, wheels and broken wheels [18], generalized θ-graphs [12], and Hertz graphs [17]. Bounds on the deficiency are known for other classes of graphs. For instance, Giaro et al. [18] provide a lower bound for Δ-regular graphs with odd n, while Giaro et al. [17] and Schwartz [42] derive an upper bound for rosettes and regular graphs, respectively. On the other hand, Petrosyan [38] shows that Eulerian multigraphs with odd $|E|$ have no interval coloring.

Only a few papers are devoted to the development of algorithms to compute the deficiency. Bouchard et al. [8] propose a tabu search algorithm to heuristically solve (MinDef). They also derive some lower bounds on the deficiency and the span of edge-colorings with minimum deficiency. They conduct computational experiments with some random graphs (with up to 1000 vertices) and some families of graphs with known deficiencies. In this body of work, the closest study to ours is the work of Altinakar et al. [1] where the authors present several IP formulations as well as a CP formulation of (MinDef). They test the performance of their models with some random graphs (with up to 100 vertices), and a complete collection of connected simple graphs with $4 \leq n \leq 8$. In their experiments, they use $K = 3n - 4$ which is a conjectured upper bound on the maximum span. They observe that the CP model is significantly better than the IP models. Later, Altinakar et al. [2] extend this work to improve the CP model by introducing a set of symmetry breaking constraints, based on graph automorphisms. However, all existing algorithms are able to solve the (MinDef) problem optimally only for very small graphs (e.g., when $n \leq 10$). Moreover, they usually fail to prove the optimality of solutions with positive deficiencies (due to weak lower bounds), and to construct solutions with small deficiencies. Our study attempts to address these drawbacks.

The remainder of this paper is organized as follows. We present the IP formulations and the solution algorithms in Section 2. We discuss the insights gained through our numerical results in Section 3 and conclude the paper in Section 4.

2 | INTEGER PROGRAMMING FORMULATIONS

Let $\mathcal{K}$ be the set of available colors and let $K = |\mathcal{K}|$. Without loss of generality, we assume that $\mathcal{K} = \{0, 1, \ldots, K-1\}$ and $K \leq |E|$. Note that the given K value might be much smaller than $|E|$ based on the considered application. For instance, in the task scheduling example discussed in the introduction, the number of job interviews (edges) could be much larger than the number of available time slots (colors). Also, since the size of our formulations depends on K, it may be advantageous to first try to find an interval coloring with a relatively small value of K and increase K only if necessary. We assume, however, that $K \geq \Delta + 1$ so that the minimum deficiency problem is always feasible, as Vizing's Theorem [45] guarantees the existence of a proper edge-coloring with $\Delta + 1$ colors.

We start with the most natural integer programming (IP) formulation of the problem, which is also presented by Altinakar et al. [1]. We define binary decision variable $x_{ijk} = 1$ if edge $\{i,j\} \in E$ is given color $k \in \mathcal{K}$, and 0 otherwise. We introduce decision variables s_i and S_i for $i \in V$ to represent the minimum and maximum color in the set of colors assigned to edges incident to vertex i, respectively. We refer to the x variables as "edge-coloring variables", while we call the s and S variables "deficiency variables". Then, the (MinDef) problem can be formulated as follows:

$$
\begin{aligned}
(\text{IP1}): \ \min \ & \sum_{i \in V} (S_i - s_i - d(i) + 1) \\
\text{s.t.} \ & \sum_{j \in N(i)} x_{ijk} \leq 1, && i \in V,\ k \in \mathcal{K}, && \text{(1a)} \\
& \sum_{k \in \mathcal{K}} x_{ijk} = 1, && \{i,j\} \in E, && \text{(1b)} \\
& S_i \geq \sum_{k \in \mathcal{K}} (k \cdot x_{ijk}), && i \in V,\ j \in N(i), && \text{(1c)} \\
& s_i \leq \sum_{k \in \mathcal{K}} (k \cdot x_{ijk}), && i \in V,\ j \in N(i), && \text{(1d)} \\
& S_i - s_i \geq d(i) - 1, && i \in V, && \text{(1e)} \\
& x_{ijk} \in \{0,1\}, && \{i,j\} \in E,\ k \in \mathcal{K}, && \text{(1f)} \\
& 0 \leq s_i \leq K - d(i), && i \in V, && \text{(1g)} \\
& d(i) - 1 \leq S_i \leq K - 1, && i \in V. && \text{(1h)}
\end{aligned}
$$

The objective function minimizes the sum of the deficiencies over all vertices, which is the definition of minimum deficiency. Constraints (1a) guarantee that adjacent edges take different colors. Constraints (1b) ensure that every edge takes exactly one color. These two sets of constraints enforce that the solution defines a proper edge-coloring. Constraints (1c) enforce S_i to be greater than or equal to the maximum color assigned to the edges incident to vertex i. Similarly, constraints (1d) enforce s_i to be less than or equal to the minimum color assigned to the edges incident to vertex i. Constraints (1e) are valid because at least $d(i)$ colors must be used to color the edges incident to vertex i. The remaining constraints provide variable bounds. Note that as x variables are binary-valued, the constraints together with the objective function force s and S variables to take on integer values. As such, s and S variables can be taken as continuous variables in the formulation.

A major problem with the above formulation is that its linear programming (LP) relaxation, which we denote as (LP1), is very weak. In fact, we have the following observation.

Proposition 1. *The optimal objective value of (LP1) is zero.*

Proof. Let $\hat{x}_{ijk} = 1/K$ for all $\{i,j\} \in E$ and $k \in \mathcal{K}$. Also, let

$$
\hat{s}_i = \begin{cases} (K-1)/2, & \text{if } K \geq 2d(i) - 1 \\ K - d(i), & \text{o.w.} \end{cases}
$$

and $\hat{S}_i = \hat{s}_i + d(i) - 1$ for all $i \in V$. Then, it is easy to see that $(\hat{x}, \hat{s}, \hat{S})$ is feasible to (LP1). Constraints (1a), (1b), and (1f) hold at $\hat{x}$ by construction. Note that (1c) and (1d) reduce to $\hat{s}_i \leq (K-1)/2 \leq \hat{S}_i,\ i \in V$. Also, by the assumption $K \geq \Delta + 1$, we have $K \geq d(i) + 1$. Thus, (1c), (1d), (1g), and (1h) are satisfied by $(\hat{s}, \hat{S})$. Finally, (1e) is tight at $(\hat{s}, \hat{S})$, so the objective value is zero. As the objective value is always nonnegative, $(\hat{x}, \hat{s}, \hat{S})$ is an optimal solution and zero is the optimal objective value of (LP1). ∎

When (IP1) is solved via a commercial solver, the branch-and-bound algorithm might take a prohibitively long time to converge because weak relaxation bounds lead to a large number of nodes in the branch-and-bound tree. Therefore, we next develop and analyze alternative formulations of the problem with tighter LP relaxations.

2.1 | Improved model of deficiencies

We propose to replace the deficiency variables in (IP1) with a new type of variables to obtain tighter LP relaxations. First, for each $i \in V$, we define the following set of intervals:

$$\mathcal{I}^i = \{[\ell, u] \, : \, \ell, u \in \mathbb{Z}, \; 0 \leq \ell, u \leq K - 1, \; u - \ell \geq d(i) - 1\} .$$

Each interval $[\ell, u] \in \mathcal{I}^i$ represents a possible pair of minimum color ℓ and maximum color u for the set of edges incident to vertex i. Also, for each $k \in \mathcal{K}$, we represent the set of intervals for vertex $i \in V$ which include color k by

$$\mathcal{I}^i(k) = \left\{[\ell, u] \in \mathcal{I}^i \, : \, k \in [\ell, u]\right\} .$$

Then, we define the binary variables

$$y_{i,[\ell,u]} = \begin{cases} 1, & \text{if} \quad \text{interval } [\ell, u] \text{ is chosen for vertex } i \\ 0, & \text{o.w.} \end{cases} , \quad i \in V, \; [\ell, u] \in \mathcal{I}^i.$$

Using y variables as the deficiency variables, we formulate the problem as:

$$\begin{aligned}
\text{(IP2)} : \; \min \; & \sum_{i \in V} \sum_{[\ell,u] \in \mathcal{I}^i} (u - \ell - d(i) + 1) \, y_{i,[\ell,u]} & & \\
\text{s.t.} \; & \sum_{k \in \mathcal{K}} x_{ijk} = 1, & \{i,j\} \in E, & \quad \text{(2a)} \\
& \sum_{[\ell,u] \in \mathcal{I}^i} y_{i,[\ell,u]} = 1, & i \in V, & \quad \text{(2b)} \\
& \sum_{j \in N(i)} x_{ijk} \leq \sum_{[\ell,u] \in \mathcal{I}^i(k)} y_{i,[\ell,u]}, & i \in V, \; k \in \mathcal{K}, & \quad \text{(2c)} \\
& x_{ijk} \in \{0, 1\}, & \{i,j\} \in E, \; k \in \mathcal{K}, & \quad \text{(2d)} \\
& y_{i,[\ell,u]} \in \{0, 1\}, & i \in V, \; [\ell, u] \in \mathcal{I}^i. & \quad \text{(2e)}
\end{aligned}$$

Constraints (2a) ensure that each edge takes exactly one color, while constraints (2b) choose exactly one interval (from the set of eligible intervals) for each vertex. Then, constraints (2c) guarantee that not only adjacent edges take different colors, but also edges incident to a vertex take colors from the set of colors included in the interval chosen for that vertex. The objective function minimizes the sum of deficiencies of the chosen intervals over all vertices. Therefore, in any optimal solution of this model, for each vertex, the minimum and the maximum color in its chosen interval must be used for an incident edge. The formal proof of (IP2) being a formulation of (MinDef) is provided in Proposition 1 in Appendix.

Next, we show that the LP relaxation of (IP2), denoted as (LP2), provides tighter bounds than (LP1).

Proposition 2. *Let ν_1^* and ν_2^* be the optimal objective values of (LP1) and (LP2), respectively. Then, $\nu_1^* \leq \nu_2^*$. Moreover, there exists an instance with $\nu_1^* < \nu_2^*$.*

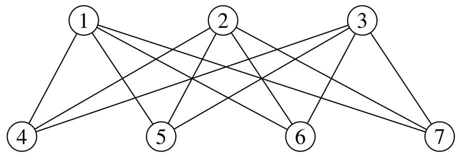


FIGURE 3 The complete bipartite graph $K_{3,4}$

Proof. As the problem is assumed to be feasible, ν_2^* is finite and always nonnegative by the definition of y variables. Then, as $\nu_1^* = 0$ by Proposition 1, we have $\nu_1^* \leq \nu_2^*$. Moreover, this inequality can be strict. In Figure 3, we provide an example for which $\nu_1^* = 0$, while $\nu_2^* = 1$ when $K = \Delta + 1 = 5$. ∎

2.2 | Improved model of edge-coloring

A set-partitioning formulation for the edge-coloring problem is introduced by Nemhauser and Park [34]. This formulation finds a minimum cardinality covering of the edges by (maximal) matchings, as the set of edges having the same color forms a matching. This so-called *matching-based* formulation is usually preferred over the edge-based formulation of the edge-coloring problem for two reasons. First, it does not have the symmetry issue of the edge-based formulation, and, second, it provides stronger relaxation bounds. With this motivation, in Section 2.2.1, we first obtain an enhanced version of the edge-based formulation by adding a specific class of matching-based valid inequalities to (IP2), and then propose a *branch-and-cut* framework to solve our model. Next, in Section 2.2.2, we introduce the so-called *matching variables* to obtain a matching-based formulation of (IP1). Finally, we describe a *column generation* based algorithm to solve the LP relaxation of the matching-based formulation.

2.2.1 | Valid inequalities

A *matching* is defined as a set of pairwise nonadjacent edges. We denote the set of all matchings in the graph G by $\mathcal{M}$. Let "conv" denote the convex hull operator. *The matching polytope of G* is the set defined as

$$P_M(G) = \text{conv}\left\{\chi^m \in \mathbb{R}^{|E|} : m \in \mathcal{M}\right\}, \tag{3}$$

where χ^m is the incidence vector of matching $m \in \mathcal{M}$, that is, $\chi_e^m = 1$ if $e \in m$ and 0 otherwise for $e \in E$. Let $\delta(i)$ denote the set of edges incident to vertex $i \in V$. *The fractional matching polytope of G* is the set

$$P_{FM} = \left\{x \in \mathbb{R}_+^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1,\ i \in V\right\}.$$

It is well known that any vector x of the matching polytope satisfies the *blossom inequalities*

$$\sum_{e \in E(S)} x_e \leq \frac{|S|-1}{2}, \quad S \subseteq V,\ |S| \text{ odd},$$

where $E(S) := \{\{i,j\} \in E : i,j \in S\}$. Moreover, adding the blossom inequalities to the fractional matching polytope is sufficient to describe the matching polytope.

Theorem 3 ([11]). *Let $P_M(G)$ be the matching polytope of graph G as defined in (3). Then,*

$$P_M(G) = \left\{x \in \mathbb{R}_+^{|E|} : \sum_{e \in \delta(i)} x_e \leq 1, i \in V, \quad \text{and} \sum_{e \in E(S)} x_e \leq (|S|-1)/2, \quad S \subseteq V \text{ s.t. } |S| \ \text{ odd}\right\}.$$

We make the following observation about the feasible solutions of (IP2). As the set of edges taking the same color must form a matching in any feasible solution, for any fixed $k \in \mathcal{K}$, the vector consisting of the variables $\left\{x_{ijk}\right\}_{\{i,j\} \in E}$ must belong to the matching polytope. Therefore, for each color $k \in \mathcal{K}$, we may add the blossom inequalities to (IP2), leading to the following enhanced formulation:

$$
\begin{aligned}
\text{(IP2-B)}:\ \min \ & \sum_{i \in V} \sum_{[\ell,u] \in \mathcal{I}^i} (u - \ell - d(i) + 1)\, y_{i,[\ell,u]} \\
\text{s.t. } & (2a)\text{– } (2e), \\
& \sum_{\{i,j\} \in E(S)} x_{ijk} \le \frac{|S|-1}{2}, \quad S \subseteq V,\ |S| \text{ odd},\ k \in \mathcal{K}.
\end{aligned} \tag{4}
$$

We denote the LP relaxation of (IP2-B) by (LP2-B).

Proposition 4. *Let ν_2^* and ν_{2B}^* be the optimal objective values of (LP2) and (LP2-B), respectively. Then, $\nu_2^* \le \nu_{2B}^*$. Moreover, there exists an instance with $\nu_2^* < \nu_{2B}^*$.*

Proof. As (IP2-B) is obtained by adding some valid inequalities to (IP2), $\nu_2^* \le \nu_{2B}^*$ trivially holds. The smallest example showing that this inequality can be strict is the complete graph with three vertices, that is, a triangle. For instance, for the triangle with $K = 3$, we obtain $\nu_2^* = 0$, whereas $\nu_{2B}^* = 1$. ∎

For the triangle example used in the above proof, among all three blossom inequalities, it is actually sufficient to add only the one corresponding to $S = V$ and $k = 1$ to (LP2) to get an optimal integer solution. This observation holds in general. Even though there are exponentially many blossom inequalities, most of them will not be binding in an optimal solution. For large instances, it is computationally impossible to add all blossom inequalities to the formulation a priori. Therefore, such inequalities should instead be added to the formulation as needed during the branch-and-bound algorithm to cut off optimal solutions of the LP relaxation. This yields a *branch-and-cut algorithm*, where a *separation problem* can be solved to identify violated blossom inequalities whenever a fractional LP solution is obtained at the nodes of the branch-and-bound tree. At any fractional node, either violated inequalities are added to the LP to cut off the current solution and the LP is re-solved, or branching is performed (i.e., no violated inequalities are found). More implementation details about our branch-and-cut algorithm are provided in Section 3.

Blossom inequalities can be separated in polynomial time. The first separation algorithm is devised by Padberg and Rao [35], whose running time is improved for dense graphs by Grötschel and Holland [19]. Later, a simpler and faster polynomial algorithm is proposed by Letchford et al. [32]. In our implementation, we use Padberg and Rao's algorithm for separation of inequalities for the *perfect matching polytope*. The perfect matching polytope of a graph $\tilde{G} = (\tilde{V}, \tilde{E})$ can be fully described as

$$
P_{PM}(\tilde{G}) = \left\{ x \in \mathbb{R}_+^{|\tilde{E}|} : \sum_{e \in \delta(i)} x_e = 1,\ i \in \tilde{V},\ \text{and} \sum_{e \in \delta(S)} x_e \ge 1,\ S \subseteq \tilde{V} \text{ s.t. } |S| \text{ odd} \right\},
$$

where $\delta(S) := \left\{ \{i,j\} \in \tilde{E} : i \in S,\ j \notin S \right\}$. The last set of inequalities in the description of $P_{PM}(\tilde{G})$ are called the *odd-set inequalities*. Padberg and Rao [35] show that the odd-set inequalities can be separated in strongly polynomial time according to the following steps. (1) Construct a *Gomory-Hu tree T* based on the given fractional solution to be separated. (2) For each edge e in the Gomory-Hu tree, check whether it is odd (even), where an edge e is odd (even) in the tree if both components of $T - e$ have an odd (even) number of vertices. (3) Find a minimum capacity odd edge $\hat{e}$ in the Gomory-Hu tree. If its capacity is less than one, return the odd-set inequality constructed by using the set of vertices in one component of $T - \hat{e}$ as the set S. This procedure returns a most violated odd-set inequality if there exists any.

Blossom inequalities can be separated for the matching polytope of G by separating the odd-set inequalities for the perfect matching polytope of a larger graph $\tilde{G} = (\tilde{V}, \tilde{E})$ which is constructed by first creating a copy $G' = (V', E')$ of $G = (V, E)$, then setting $\tilde{V} = V \cup V'$ and $\tilde{E} = E \cup E' \cup \{(i, i') : i \in V\}$. The separation procedure works by using the following reduction between the two polytopes [41]: If we take $\hat{x} \in P_M(G)$ and construct

$$
\tilde{x}_{\tilde{e}} = \begin{cases} \hat{x}_e, & \text{if } \ \tilde{e} = e \in E \\ \hat{x}_e, & \text{if } \ \tilde{e} = e' \in E' \\ 1 - \sum_{e \in \delta(i)} \hat{x}_e, & \text{if } \ \tilde{e} = (i, i') \end{cases}, \quad \tilde{e} \in \tilde{E},
$$

then $\tilde{x} \in P_{PM}(\tilde{G})$. Using this relation, a blossom inequality which is valid for $P_M(G)$ can be converted to an odd-set inequality which is valid for $P_{PM}(\tilde{G})$, and vice versa.

2.2.2 | Matching-based formulation

Next, we propose a matching-based formulation of (MinDef). We first define parameters

$$a_{m,e} = \begin{cases} 1, & \text{if } e \in m \\ 0, & \text{o.w.} \end{cases}, \quad m \in \mathcal{M},\ e \in E,$$

and

$$b_{m,i} = \begin{cases} 1, & \text{if } \exists\, e = \{i,j\} \in E \text{ s.t. } e \in m \\ 0, & \text{o.w.} \end{cases}, \quad m \in \mathcal{M},\ i \in V.$$

Note that these parameters have the following relationship:

$$b_{m,i} = \sum_{e=\{i,j\}: j \in N(i)} a_{m,e}, \quad m \in \mathcal{M},\ i \in V. \tag{5}$$

Next, we introduce the binary decision variables, referred to as the *matching variables*,

$$z_{k,m} = \begin{cases} 1, & \text{if } m \text{ is chosen and given color } k \\ 0, & \text{o.w.} \end{cases}, \quad k \in \mathcal{K},\ m \in \mathcal{M},$$

and replace the edge-coloring variables with the matching variables in (IP2) using the relation

$$x_{ijk} = \sum_{m \in \mathcal{M}: e \in m} z_{k,m} = \sum_{m \in \mathcal{M}} a_{m,e} z_{k,m}, \quad e = \{i,j\} \in E,\ k \in \mathcal{K}. \tag{6}$$

We thus obtain the following formulation of (MinDef):

$$\text{(IP3)}: \ \min \sum_{i \in V} \sum_{[\ell,u] \in \mathcal{I}^i} (u - \ell - d(i) + 1)\, y_{i,[\ell,u]}$$

$$\text{s.t.} \sum_{[\ell,u] \in \mathcal{I}^i} y_{i,[\ell,u]} = 1, \qquad i \in V, \tag{7a}$$

$$\sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} a_{m,e} z_{k,m} = 1, \qquad e \in E, \tag{7b}$$

$$\sum_{m \in \mathcal{M}} z_{k,m} \le 1, \qquad k \in \mathcal{K}, \tag{7c}$$

$$\sum_{m \in \mathcal{M}} b_{m,i} z_{k,m} \le \sum_{[\ell,u] \in \mathcal{I}^i(k)} y_{i,[\ell,u]}, \qquad i \in V,\ k \in \mathcal{K}, \tag{7d}$$

$$z_{k,m} \in \{0,1\}, \qquad k \in \mathcal{K},\ m \in \mathcal{M}, \tag{7e}$$

$$y_{i,[\ell,u]} \in \{0,1\}, \qquad i \in V,\ [\ell,u] \in \mathcal{I}^i. \tag{7f}$$

(IP3) partitions the edges into disjoint matchings, assigns different colors to these matchings and colors all the edges in a matching by the matching's color. Constraints (7b) enforce that each edge is covered by exactly one matching, which takes exactly one color, while constraints (7c) assign each color to at most one matching. Therefore, constraints (7d) guarantee that the edges incident to a vertex take different colors. The formal proof that (IP3) is a formulation of (MinDef) is provided in Proposition 2 in Appendix.

We denote the LP relaxation of (IP3) by (LP3) and represent its optimal objective value by ν_3^*.

Proposition 5. *(LP3) and (LP2-B) are equivalent, that is,* $\nu_3^* = \nu_{2B}^*$.

Proof. Let $(\hat{z}, \hat{y})$ be a feasible solution of (LP3), and let $\hat{x}$ be the vector constructed from $\hat{z}$ using (6). Then, it is straightforward to verify that $(\hat{x}, \hat{y})$ satisfy (2a)-(2c) and the LP relaxation of (2d)-(2e). Next, for each $k \in \mathcal{K}$, (7c) implies $\sum_{m\in\mathcal{M}} \hat{z}_{k,m} \leq 1$, and hence by (6), the vector $\left\{\hat{x}_{ijk}\right\}_{\{i,j\}\in E}$ is a convex combination of matching incidence vectors χ^m, $m \in \mathcal{M}$ (where the empty matching has weight $1 - \sum_{m\in\mathcal{M}} \hat{z}_{k,m}$). Thus, for each k, the vector $\left\{\hat{x}_{ijk}\right\}_{\{i,j\}\in E}$ belongs to the matching polytope $P_M(G)$, and hence Theorem 3 implies that $(\hat{x}, \hat{y})$ satisfies the blossom inequalities (4), and thus is feasible to (LP2-B). As the objective functions of the two models are the same, we obtain $\nu_3^* \geq \nu_{2B}^*$.

Now, let $(\tilde{x}, \tilde{y})$ be a feasible solution of (LP2-B). As $\tilde{x}$ satisfies all the blossom inequalities, due to Theorem 3, we know that for each $k \in \mathcal{K}$, the vector $\left\{\tilde{x}_{ijk}\right\}_{\{i,j\}\in E}$ belongs to $P_M(G)$, thus can be written as a convex combination of the incidence vectors of the matchings in $\mathcal{M}$, based on the definition of $P_M(G)$ given in (3). For each $k \in \mathcal{K}$, let $\lambda^k \in \mathbb{R}_+^{|\mathcal{M}|}$ be a vector satisfying

$$\left\{\tilde{x}_{ijk}\right\}_{\{i,j\}\in E} = \sum_{m\in\mathcal{M}} \lambda_m^k \chi^m \quad \text{and} \quad \sum_{m\in\mathcal{M}} \lambda_m^k = 1.$$

Then, define $\tilde{z}$ by $\tilde{z}_{k,m} = \lambda_m^k$ for all $k \in \mathcal{K}$, $m \in \mathcal{M}$. It is straightforward to verify that (6) is satisfied, which in turn implies $(\tilde{z}, \tilde{y})$ satisfies (7b) and (7d), and hence is a feasible solution of (LP3). As this solution has the same objective value in (LP3) as $(\tilde{x}, \tilde{y})$ has in (LP2-B), this implies $\nu_3^* \leq \nu_{2B}^*$. ∎

This result combined with Propositions 2 and 4 implies that

$$\nu_1^* \leq \nu_2^* \leq \nu_3^* = \nu_{2B}^*.$$

Therefore, (IP3) has the strongest LP relaxation among the formulations presented so far. Even though (LP3) is tighter than (LP2) in general, we show that they have the same strength for bipartite graphs. We first review a result about matchings which is the basis for this claim.

Theorem 6 ([41]). *If G is bipartite, then* $P_M(G) = P_{FM}(G)$.

This leads to the following result.

Proposition 7. *If the graph G is bipartite, then (LP3) and (LP2) are equivalent, that is,* $\nu_3^* = \nu_2^*$.

Proof. Propositions 4 and 5 imply that $\nu_2^* \leq \nu_3^* = \nu_{2B}^*$. Thus, it is sufficient to show that $\nu_{2B}^* \leq \nu_2^*$. Let $(\hat{x}, \hat{y})$ be an optimal solution of (LP2). For $k \in \mathcal{K}$, we define the vector $\tilde{x}^k \in \mathbb{R}^{|E|}$ such that $\tilde{x}_e^k = \hat{x}_{ijk}$ for all $e = \{i,j\} \in E$. As $\tilde{x}^k \in P_{FM}(G)$ and G is bipartite, by Theorem 6, we have $\tilde{x}^k \in P_M(G)$. Then, Theorem 3 implies that $\hat{x}$ satisfies all blossom inequalities. Therefore, $(\hat{x}, \hat{y})$ is a feasible solution of (LP2-B), which shows that $\nu_{2B}^* \leq \nu_2^*$. ∎

The number of z variables in (IP3) grows exponentially in the size of the graph, and thus for moderate size graphs, it is not possible to enumerate all matchings in the graph, and hence explicitly construct the formulation (IP3) in a reasonable amount of time. However, most of these variables will take value zero in an optimal solution. Therefore, a possible solution approach is to apply a *branch-and-price* algorithm [7]. The first requirement of branch-and-price is to solve the LP relaxation of the model via *column generation* [9].

Column generation solves an LP by solving a restricted version of the LP, known as the *restricted master LP*, in which most decision variables of the original LP are not included (they are implicitly restricted to have value 0). Decision variables are then iteratively added to the restricted master LP until it yields a solution to the original LP. Adding a decision variable to the restricted master LP is referred to as "column generation" because this is implemented by adding a column to the constraint matrix of the LP. At every step of the column generation algorithm, the current restricted master LP is solved, its optimal dual solution values are passed to the so-called *pricing problem*, new columns are generated and added to the restricted master LP, and then the restricted master LP is re-solved. This loop is repeated until no more columns are generated.

We first explain how to generate new columns for the restricted master LP, assuming it has an optimal solution. In our derivation, we exclude the upper bounds on the x and y variables in the relaxation of constraints (7e) and (7f) as they are implied by the other constraints in (LP3). We denote the dual variables associated with the constraints (7a)-(7d) of the restricted version of (LP3) by θ, γ, δ, and Ω, respectively. Given an optimal dual solution $(\theta^*, \gamma^*, \delta^*, \Omega^*)$ of the restricted master LP, the reduced cost of variable $z_{k,m}$ is

$$c^*_{k,m} := -\delta^*_k - \sum_{e=\{i,j\}\in E} a_{m,e}(\gamma^*_e + \Omega^*_{i,k} + \Omega^*_{j,k}).$$

The pricing problem is to find

$$\min\left\{c^*_{k,m} : k \in \mathcal{K}, m \in \mathcal{M}\right\}. \tag{8}$$

To solve (8), for each $k \in \mathcal{K}$, define $g^*_{e,k} := \gamma^*_e + \Omega^*_{i,k} + \Omega^*_{j,k}$, for $e = \{i,j\} \in E$ and solve the following problem:

$$v^*_k := \max\left\{\sum_{e\in m} g^*_{e,k} : m \in \mathcal{M}\right\} \tag{9}$$

Problem (9) is a *Maximum Weight Matching* problem, and hence it is polynomially solvable, for example, by Edmond's algorithm [11]. For each $k \in \mathcal{K}$, let m^{*k} be an optimal solution of (9). By definition of problem (9),

$$r^*_k := \min\left\{c^*_{k,m} : m \in \mathcal{M}\right\} = -\delta^*_k - \sum_{e\in m^{*k}} g^*_{e,k} = -\delta^*_k - v^*_k$$

and thus, if $r^*_k < 0$, then a column with negative reduced cost is found. Specifically, the variable $z_{k,m^{*k}}$ is added to the restricted master LP. In the case that more than one column is found (from different values of k), different strategies can be used to determine the columns to be added as long as at least one column is added. In our implementation, at every iteration, we solve the pricing problem (9) for all $k \in \mathcal{K}$ and add all of the columns found that have $r^*_k < 0$. When $r^*_k \geq 0$ for all $k \in \mathcal{K}$, an optimal solution of the current restricted master LP is also optimal for (LP3).

A standard way to initialize the restricted master LP is to use an artificial column with a relatively large objective coefficient to ensure feasibility, which can be removed once the optimal value of zero is achieved (e.g., see [9]). An alternative approach is to use a known feasible LP solution in the initialization. We choose the latter as it usually provides a better guidance in the beginning of the column generation algorithm. In order to get a feasible solution of (LP3) to initialize the column generation procedure, we propose solving the LP relaxation of the *Minimum Cardinality Edge-Coloring* problem, which is

$$\begin{aligned}
\text{(EC)}:\ v^* := \min\ & \sum_{m\in\mathcal{M}} z_m \\
\text{s.t.}\ & \sum_{m\in\mathcal{M}} a_{m,e} z_m = 1, \quad e \in E, \\
& z \in \mathbb{R}^{|\mathcal{M}|}_+.
\end{aligned}$$

As (EC) has exponentially many variables, we solve this model via column generation as well. Let π denote the dual variables and let π^* be an optimal dual solution at the current iteration. The pricing problem is again a maximum weight matching problem, which is equivalent to (9) with edge weights π^* in the objective. We find an initial feasible solution for (EC) (i.e., a set of matchings) as follows. First, we find a maximum cardinality matching (via Edmond's blossom shrinking algorithm). Then, we remove these edges from the graph, find a maximum cardinality matching in the remaining graph, and repeat until all of the edges are removed. Suppose that we obtain z^* as an optimal solution of (EC), with objective value v^*, and let $\mathcal{M}^* := \left\{m \in \mathcal{M} : z^*_m > 0\right\}$. Then, we can initialize (LP3) with columns corresponding to $z_{k,m}$ for all $m \in \mathcal{M}^*$, $k \in \mathcal{K}$. The initial LP formed for (LP3) from this procedure is always feasible when $K \geq \Delta + 1$, which is proven in Proposition 3 in Appendix. In our numerical experiments, this initial phase (i.e., solving (EC) via column generation) took an insignificant amount of time (less than 0.001 seconds on average).

In order to solve (IP3) to optimality, column generation should be combined with branch-and-bound, which results in a branch-and-price algorithm. Specifically, in branch-and-price, a branch-and-bound tree search is done where a predetermined branching rule is performed on the integer variables in the problem. The search is initialized with the master LP containing no branching restrictions, which is formed by the columns in a given feasible LP relaxation solution. At each node of the branch-and-bound tree, the master LP, augmented by branching constraints, is solved via column generation. If the optimal master LP solution does not satisfy integrality constraints, then branching is performed. As we do not include the branch-and-price algorithm in our computational experiments, we do not provide further details of the algorithm and refer the reader to Barnhart et al. [7].

2.3 | Symmetry breaking

Symmetry is very common in combinatorial optimization problems, especially in graph coloring problems. Existence of symmetry in a problem indicates that a feasible solution can be permuted to obtain an equivalent feasible solution. In our problem, given an edge-coloring of the graph, we can obtain many other equivalent feasible solutions by permuting the labels of the colors. For instance, for the interval 6-coloring of W_7 given in Figure 1(a), we obtain an equivalent interval coloring if we relabel the colors 1,2,3,4,5,6 as 6,5,4,3,2,1, respectively. Symmetry provides challenges for branch-and-bound tree search algorithms because when a branching is performed to remove an infeasible solution, an equivalent symmetric solution may still be feasible.

One technique for addressing this challenge is to add inequalities that exclude symmetric solutions. There are two commonly used approaches to manage symmetry breaking inequalities: Generating *dynamic symmetry breaking inequalities* during the solution process; and adding *static symmetry breaking inequalities* to the initial formulation, cutting some of the symmetric solutions while keeping at least one optimal solution [33]. We use the latter approach to reduce the symmetry in our problem. For each $k \in \mathcal{K}$, we introduce a new binary variable w_k which takes value 1 if color k is used, and 0 otherwise. Then, we enforce the use of only consecutive colors starting from 0. In other words, we do not use color $k+1$ unless color k is used. As we need to use at least Δ different colors, we can fix the first Δ of w variables to 1.

We apply these symmetry breaking techniques to all of the formulations that we have presented. In the first model, (IP1), we replace (1a) with

$$\sum_{j \in N(i)} x_{ijk} \leq w_k, \qquad i \in V,\ k \in \mathcal{K},$$

and also add the following constraints:

$$w_k = 1, \qquad k = 0, \ldots, \Delta - 1, \tag{11}$$

$$w_k \leq w_{k-1}, \qquad k = \Delta, \ldots, K-1, \tag{12}$$

$$w_k \in \{0,1\}, \qquad k \in \mathcal{K}. \tag{13}$$

For the models (IP2) and (IP2-B), in addition to (11)-(13), we also add

$$\sum_{[\ell,u] \in \mathcal{I}^i(k)} y_{i,[\ell,u]} \leq w_k, \qquad i \in V,\ k \in \mathcal{K}.$$

Lastly, for the (IP3) model, we replace (7c) with

$$\sum_{m \in \mathcal{M}} z_{k,m} \begin{cases} = 1, & k = 0, \ldots, \Delta - 1, \\ \leq 1, & k = \Delta, \\ \leq \sum_{m \in \mathcal{M}} z_{k-1,m}, & k = \Delta+1, \ldots, K-1. \end{cases} \tag{14}$$

Note that these modifications do not change the pricing problem given in (9), but affect the criteria used to detect the new columns. Accordingly, a new column would be added to the master LP if $-v_k^* - \tilde{\delta}_k^* < 0$, where v_k^* denotes the optimal value of the problem (9) for $k \in \mathcal{K}$, whereas the value $\tilde{\delta}_k^*$ is calculated using optimal values δ^* of the dual variables associated with (14) as

$$\tilde{\delta}_k^* = \begin{cases} \delta_k^*, & \text{if } k = K-1 \text{ or } k < \Delta \\ \delta_k^* - \delta_{k+1}^*, & \text{o.w.} \end{cases}, \quad k \in \mathcal{K}.$$

Also note that the feasible solution provided in Proposition 3 (in Appendix) remains feasible for (LP3) after the addition of the symmetry breaking constraints.

Consider now a solution with integer p_e as the color assigned to the color of edge e, for $e \in E$, and let $K' = \max p_e : e \in E$. Then the solution that assigns the color $p'_e = K' - p_e$ to each edge $e \in E$ is a symmetric solution with identical objective value. To reduce the impact of this symmetry, we choose an edge $\{i,j\}$ with maximum $d(i) + d(j)$ value, and restrict the colors allowed for that edge to the set $\{0, 1, \dots, \lfloor (K-1)/2 \rfloor\}$. Note that we use K instead of K' in the upper bound of the interval because K' cannot be used as it depends on the solution.

3 | COMPUTATIONAL EXPERIMENTS

Constraint programming formulation. Constraint programming (CP) is another widely used approach to model graph coloring problems, especially for vertex-coloring. As noted by Gualandi and Malucelli [20], the combinatorial structure of coloring problems makes CP approaches often efficient and competitive with respect to the IP ones; however standard CP approaches lack efficient mechanisms to compute tight lower bounds and to guide the search toward the optimal solution. Therefore, we compare our IP models with a CP model.

We introduce decision variables c_{ij} to represent the color given to edge $\{i,j\} \in E$, and decision variables η_i to denote the deficiency of vertex $i \in V$. Also, we again use decision variables s_i and S_i to represent the minimum and maximum color in the set of colors assigned to edges incident to vertex $i \in V$, respectively. Then, a natural constraint programming formulation for (MinDef) problem follows as [1]:

$$\text{(CP)}: \min \sum_{i \in V} \eta_i$$

$$\text{s.t. } \underset{j \in N(i)}{\text{All Diff}} \; (c_{ij}), \qquad i \in V, \tag{15a}$$

$$S_i = \max_{j \in N(i)} (c_{ij}), \qquad i \in V, \tag{15b}$$

$$s_i = \min_{j \in N(i)} (c_{ij}), \qquad i \in V, \tag{15c}$$

$$\eta_i = S_i - s_i + 1 - d(i), \qquad i \in V, \tag{15d}$$

$$c_{ij} \in \{0, \dots, K-1\}, \qquad \{i,j\} \in E,$$

$$\eta_i \in \{0, \dots, K - d(i)\}, \qquad i \in V,$$

$$s_i \in \{0, \dots, K - d(i)\}, \qquad i \in V,$$

$$S_i \in \{d(i) - 1, \dots, K-1\}, \qquad i \in V.$$

The objective function minimizes the sum of the deficiencies over all vertices. Constraints (15a) use "All Different" predicate to enforce that adjacent edges take different colors. Constraints (15b), (15c), and (15d) are necessary to define S, s, and η variables, respectively. The remaining constraints provide the sets of values the variables can take. As in the IP models, we also introduce the following symmetry breaking constraints:

$$|\{i,j\} \in E : c_{ij} = k| \geq 1, \quad k = 0, \dots, \Delta - 1,$$

$$z_{i^*j^*} \leq \lfloor (K-1)/2 \rfloor,$$

where $\{i^*, j^*\}$ is an edge in E with maximum $d(i) + d(j)$ value.

TABLE 1 Deficiency of the instances when $K = \Delta + 1$

n	$D = 0.2$	$D = 0.5$	$D = 0.8$
11	$\{0, 0, 0, 0, 0\}$	$\{0, 0, 0, 1, 0\}$	$\{2, 5, 1, 1, 1\}$
15	$\{0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0\}$	$\{0, 6, 1, 3, 1\}$
19	$\{0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0\}$	$\{8, 3, 5, 7, 6\}$
20	$\{0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0\}$
23	$\{0, 0, 0, 0, 0\}$	$\{0, 1, 0, 0, 0\}$	$\{8, 12, 7, 6, 3\}$
27	$\{0, 0, 0, 0, 0\}$	$\{1, 1, 0, 0, 1\}$	$\{9, 8, 8, 6, 9\}$
30	$\{0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0\}$	$\{0, 0, 0, 0, 0\}$
31	$\{0, 0, 0, 0, 0\}$	$\{0, 3, 0, 0, 1\}$	$\{9, 15, 11, 11, 12\}$

Test instances. We perform our numerical experiments on a test data set consisting of randomly generated problem instances for which the expected edge density of the graph (measured as $D = \frac{2|E|}{n(n-1)}$) takes values 0.2, 0.5, and 0.8. The graphs are generated as Erdős-Rényi random graphs using the `gnp_random_graph` function in the NetworkX package [21]. As stated by Bouchard et al. [8], which we also observe in our numerical experiments, graphs with odd number of vertices are more challenging because they typically have larger deficiencies than similar size graphs with even number of vertices. Therefore, we analyze more instances with odd number of vertices. In particular, we consider graphs with the number of vertices $n \in \{11, 15, 19, 20, 23, 27, 30, 31\}$. For each ($D$, n) combination, we generate five different graphs (Data sets used in our tests are available online at https://sites.google.com/site/mervebodr/). In order to obtain larger deficiencies, and thus more challenging instances, we use $K = \Delta + 1$ in our experiments, unless otherwise stated.

The deficiency of our test instances (obtained by solving the problems to optimality without time limit) are given in Table 1, where the values in the curly brackets correspond to the five different instances generated for a fixed (D, n) combination. Naturally, instances with higher density have higher deficiencies. Also, as previously mentioned, the instances with even n have lower deficiencies; in particular they are all interval colorable.

Implementation details. We implement all algorithms in C++ using IBM ILOG CPLEX 12.4 for solving all LPs and IPs, IBM ILOG CP OPTIMIZER 12.4 for solving CPs, and LEMON Graph Library 1.2.3 [10] for seeking maximum weight matchings and constructing Gomory-Hu trees. We run all experiments using a single thread on a Linux workstation with 3.16 GHz Intel Xeon CPUs and 8 GB memory. For all runs, we impose a solution time limit of one hour.

In order to solve (IP2-B), we embed the generation of blossom inequalities within a branch-and-bound algorithm, leading to a branch-and-cut algorithm. However, before starting the branch-and-cut algorithm, we first solve the LP relaxation of (IP2-B), that is, (LP2-B), via a cutting-plane algorithm. We start the cutting-plane algorithm by solving (LP2). At every iteration, we solve the LP, generate the blossom cuts among which we add the ones violated by the current LP solution to the LP. For each available color, we only add the most violated blossom cut, if there exists any. The cutting-plane algorithm stops when no more violated blossom cuts are found. In order to limit the number of cuts added at this phase, once the LP solve is done, we remove all the cuts that are not tight at the optimal LP solution. The purpose of this first phase implementation is that CPLEX can generate its own cuts based on the constraints in the given model formulation, so it can generate more and/or stronger cuts by using more information about the problem. Then, in the second phase, we apply the branch-and-cut algorithm where the addition of blossom cuts is implemented within a *UserConstraintCallback* in CPLEX to cut off fractional solutions. Symmetry breaking constraints, as described in Section 2.3, are used in all the experiments.

3.1 | Comparison of LP relaxations

We first compare the computational performance of solving the LP relaxations of (IP2-B) and (IP3). Although the optimal values of (LP2-B) and (LP3) are equal, as shown in Proposition 5, the solution times of these models might affect the decision of which model to implement. We present the results of this experiment for $K = \Delta + 1$ and $K = \Delta + 1 + \text{def}$ in Table 2, where def is the deficiency of the former case. As the number of instances that are not interval colorable when $K = \Delta + 1 + \text{def}$ is very small for $D = 0.2$ and $D = 0.5$, we only consider the instances with $D = 0.8$. We provide the result only for three values of n, which involve the instances whose LP relaxations take the longest time to solve.

TABLE 2 Some statistics about solving the LP relaxation

			(LP2-B)			(LP3)		
K	D	n	Cuts	Iters	Time	Cols	Iters	Time
$\Delta+1$	0.2	23	26.0	4.0	0.1	308.4	5.2	0.2
		27	30.6	5.0	0.2	406.6	29.8	0.5
		31	26.2	4.2	0.3	622.6	47.8	1.8
	0.5	23	295.2	12.4	3.4	1922.4	60.8	4.9
		27	722.8	20.6	22.4	3290.2	51.2	20.5
		31	1094.4	27.0	73.2	5286.4	61.2	77.2
	0.8	23	1121.4	20.4	31.9	5187.6	33.0	20.9
		27	1571.0	25.0	104.7	8369.0	54.4	199.0
		31	2357.2	30.6	348.4	12625.2	71.2	276.8
$\Delta+1+\text{def}$	0.8	23	884.4	40.6	161.3	6459.8	18.6	47.5
		27	992.6	39.4	337.5	10340.0	33.6	327.0
		31	1615.2	61.2	1688.6	16489.8	41.0	1624.8

(LP2-B) is solved via a cutting plane algorithm (as explained in the Implementation details section), while column generation is used to solve (LP3). As such, for the former, we report the number of blossom cuts added to the LP ("Cuts"), the number of times the LP is solved ("Iters") and the total time spent in seconds ("Time"), whereas for the latter, we report the number of columns added to the restricted master LP ("Cols"), the number of times the restricted master LP is solved ("Iters") and the total time spent in seconds ("Time"). All the values correspond to the averages, that is, each row of the table shows the averages over five instances.

These results indicate that the solution times for (LP2-B) and (LP3) are comparable. Although it appears that (LP3) might be more efficient for larger and denser instances, the time savings are not very significant for our test instances. Therefore, we do not implement a branch-and-price algorithm to test (IP3) in our numerical experiments.

3.2 | Comparison of alternative approaches

We next compare four different formulations, namely (IP1), (IP2), (IP2-B), and (CP), in terms of solution time and optimality gaps obtained at the end of the given time limit. Table 3 presents results about the average solution times, and absolute optimality gaps for instances that are not solved within the time limit. Each row corresponds to a different combination of density D and number of nodes n. For solution times, the averages are taken over five instances having that combination. In computing these averages, one hour is used for instances that were not solved within the time limit. For absolute gaps, Table 3 reports first the number of unsolved instances in parentheses (when positive), followed by the minimum and maximum absolute gap among the unsolved instances.

We observe that the problem difficulty increases with D, in the sense that the methods fail more often in solving instances to optimality and end up with larger optimality gaps as D increases. As such, all the formulations solve all instances with $D=0.2$ to optimality, whereas (IP2-B) clearly outperforms the others for the cases with $D=0.5$ and $D=0.8$, solving all of the instances except one. This is mainly because the instances with larger densities have larger deficiencies (see Table 1), and for such instances the strength of the relaxation bound becomes more important. In terms of solution time, we find that (CP) is significantly faster than the other methods for the instances with $D = 0.2$. On the other hand, when $D=0.8$, (IP2-B) solves far more instances than any other method, and has significantly smaller solution times. For the middle density level $D=0.5$, (IP2-B) solves many more instances than the alternative methods, and has faster solution times on average, although there are individual instances at this level for which (CP) is faster than (IP2-B). We also found that all of the instances that (CP) can solve within the time limit have zero deficiency.

In Table 4, we provide some statistics about the number of nodes in the branch-and-bound tree for our IP models, including the percentage of the instances solved at the root node, the median of the number of nodes explored in the tree, and the percentage of the instances where more than 1000 nodes are processed, in the columns labeled as "At Root", "Median" and "$\geq$ 1000 ", respectively. We see that (IP2-B) solves more than 85% of the instances at the root node, which shows the strength of its LP relaxation. Overall, (IP2-B) explores very few nodes in the branch-and-bound tree, whereas (IP1) and (IP2) usually lead to large

TABLE 3 Solution times and ranges of absolute optimality gaps of the models for different density levels and number of nodes

		Average Solution Times (seconds)				(Number unsolved): Min Gap, Max Gap			
D	*n*	(IP1)	(IP2)	(IP2-B)	(CP)	(IP1)	(IP2)	(IP2-B)	(CP)
0.2	11	0.0	0.0	0.0	0.0				
	15	0.2	0.1	0.1	0.0				
	19	0.6	0.2	0.1	0.0				
	20	0.7	0.6	0.5	0.0				
	23	16.6	0.9	0.4	0.1				
	27	79.7	7.7	11.0	0.4				
	30	371.3	16.2	9.4	1.6				
	31	1444.5	1025.1	128.0	2.7	(1):3,3			
0.5	11	720.2	0.1	0.1	720.0	(1):1,1			(1):1,1
	15	889.8	26.6	1.3	0.5				
	19	2609.4	1270.8	2.9	4.0	(2):1,2			
	20	3256.5	1560.5	21.1	3.7	(3):2,3			
	23	3600.0	2934.6	62.2	741.0	(5):2,6	(2):1,1		(1):1,1
	27	3600.0	3538.1	745.1	2242.4	(5):1,8	(4):3,9	(1):2,2	(3):1,3
	30	3600.0	3142.5	119.6	2053.3	(5):3,15	(4):4,6		(2):1,1
	31	3600.0	3600.0	347.4	3600.0	(5):7,14	(5):2,14		(5):1,5
0.8	11	3600.0	2711.5	0.3	3600.0	(5):1,5	(5):1,7		(5):1,5
	15	3600.0	3449.5	14.3	2882.9	(5):1,7	(4):1,3		(4):1,6
	19	3600.0	3533.7	16.5	3600.0	(5):6,9	(4):1,8		(5):3,8
	20	3600.0	960.3	7.9	19.2	(5):1,3			
	23	3600.0	3600.0	42.7	3600.0	(5):11,15	(5):5,14		(5):4,12
	27	3600.0	3600.0	241.9	3600.0	(5):13,17	(5):13,∞		(5):8,10
	30	3600.0	3172.3	358.5	3309.4	(5):3,18	(4):9,∞		(4):1,7
	31	3600.0	3600.0	1314.2	3600.0	(5):14,26	(5):∞, ∞		(5):14,16

TABLE 4 The number of node statistics for the IP formulations

D	Model	At Root	Median	$\geq$ 1000
0.2	(IP1)	10.0	2316	57.5
	(IP2)	42.5	12	20.0
	(IP2-B)	85.0	0	7.5
0.5	(IP1)	2.5	448837	92.5
	(IP2)	12.5	148611	80.0
	(IP2-B)	90.0	0	0.0
0.8	(IP1)	0.0	325550	100.0
	(IP2)	0.0	277278	100.0
	(IP2-B)	87.5	0	0.0

branch-and-bound trees. As (IP2-B) performs significantly better than (IP1) and (IP2), we do not consider (IP1) and (IP2) in the remaining experiments.

3.3 | Impact of the number of allowed colors

In our final experiment, we investigate how the deficiencies change as we increase the number of allowed colors, K. For this aim, we pick the instances from our test set that are not interval colorable when $K = \Delta + 1$, and use (IP2-B) and (CP) to find

TABLE 5 Impact of K on deficiency for instances with $D = 0.5$

	n	|*E*|	Δ	*K*	LB	UB	CP	IP time	CP time
graph_D05_1	11	30	7	8	1	1	**1**	0	3600
				9	0	0	0	0	1
graph_D05_2	23	117	13	14	1	1	**1**	3	3600
				15	0	0	0	4	13
graph_D05_3	27	174	16	17	1	1	**1**	34	3600
				18	0	0	**1**	38	3600
graph_D05_4	27	177	17	18	**0**	**2**	**1**	3600	3600
				20,19	0	0	**1**	493	3600
graph_D05_5	27	177	18	19	1	1	**3**	58	3600
				20,22	0	0	**3**	99	3600
graph_D05_6	31	232	19	20	3	3	**5**	335	3600
				23,25	**0**	**3**	**5**	3600	3600
graph_D05_7	31	247	20	21	1	1	**4**	165	3600
				22,25	0	0	**4**	513	3600

the deficiency values of these instances for some increased values of K. Note that the number of instances (out of 40) that are interval colorable are 40, 33 and 11 for $D = 0.2$, $D = 0.5$, and $D = 0.8$, respectively.

We experiment with increasing the value of K gradually as follows: We first solve a problem with $K = \Delta + 1$ colors (with the time limit of one hour), then increase K by the upper bound obtained on the deficiency when $K = \Delta + 1$. This helps us to use some information from the previous solve. Specifically, we do the following:

- For (CP): We solve the problem with $K = \Delta + 1$. Let " def_{CP} " denote the deficiency of the best solution found within the time limit. Then, we solve the problem with $K = \Delta + 1 + \text{def}_{\text{CP}}$, where we provide the best feasible solution of the previous solve as a starting solution to the CP solver.
- For (IP2-B): We solve the problem with $K = \Delta + 1$. Let " $\text{def}_{\text{IP2-B}}$ " denote the deficiency of the best solution found within the time limit. Then, we solve the problem with $K = \Delta + 1 + \text{def}_{\text{IP2-B}}$. As in the CP case, we provide the best solution found in the previous solve as an initial solution to the IP solver. In addition, we reuse some blossom cuts generated in the previous solve to tighten the model. We choose to reuse some of the cuts that are generated when solving the LP relaxation of the previous model; the ones that are tight at the optimal LP solution.

Table 5 includes all seven instances with $D = 0.5$ that are not interval colorable with $K = \Delta + 1$ colors. The columns labeled as "LB" and "UB" correspond to the lower and upper bound values on the deficiency reported by (IP2-B), respectively, while the "CP" column refers to the upper bounds on the deficiency found by (CP). The solution times in seconds are given in the last two columns. For each instance, in the "K" column, the first value is equal to $\Delta + 1$, whereas the second line provides "$\Delta + 1 + \text{def}_{\text{IP2-B}}, \Delta+1 + \text{def}_{\text{CP}}$" (a single value is given if they are equal). The bounds for the instances that could not be solved within the time limit are shown in bold.

We observe that all the instances except one (which could not be solved) become interval colorable after adding only a few colors. Moreover, (IP2-B) is able to find an interval coloring for those instances in a reasonable amount of time, while (CP) fails in the majority of them.

As seen in Table 5, for graph_D05_6, none of the methods made any improvement on the upper bound, and the lower bound is zero. For $D = 0.8$, we do not report such "no improvement" instances. Table 6 illustrates the results for 18 of 29 instances with $D = 0.8$ that are not interval colorable when $K = \Delta + 1$.

(IP2-B) proves that nine instances (the unhighlighted ones in the table) become interval colorable after allowing $\text{def}_{\text{IP2-B}}$ more colors. Note that $\text{def}_{\text{IP2-B}}$ is exact in all of the instances. Moreover, (IP2-B) proves that in three instances (the dark highlighted ones in the table), $\text{def}_{\text{IP2-B}}$ additional colors are not sufficient to obtain an interval coloring. The light highlighted instances in the table could not be solved to optimality, although their upper bounds have been improved by either of the two methods. Finally, we remark that (IP2-B) performs significantly better than (CP) in terms of solvability and especially finding

TABLE 6 Impact of K on deficiency for instances with $D = 0.8$

	n	$\|E\|$	Δ	K	LB	UB	CP	IP time	CP time
graph_D08_1	11	40	9	10	1	1	**1**	0	3600
				11	0	0	0	0	3
graph_D08_2	11	45	10	11	2	2	**2**	0	3600
				12	1	1	**1**	7	3600
graph_D08_3	11	45	10	11	1	1	**1**	0	3600
				12	1	1	**1**	387	3600
graph_D08_4	11	48	10	11	5	5	**5**	0	3600
				16	**0**	**1**	**1**	3600	3600
graph_D08_5	15	81	14	15	1	1	**1**	2	3600
				16	0	0	0	4	425
graph_D08_6	15	82	13	14	3	3	**3**	1	3600
				17	1	1	**1**	481	3600
graph_D08_7	15	92	14	15	6	6	**6**	1	3600
				21	**0**	**6**	**3**	3600	3600
graph_D08_8	19	129	16	17	5	5	**5**	8	3600
				22	0	0	**5**	80	3600
graph_D08_9	19	142	17	18	8	8	**8**	8	3600
				26	**0**	**8**	**7**	3600	3600
graph_D08_10	19	142	18	19	7	7	**7**	10	3600
				26	**0**	**7**	**6**	3600	3600
graph_D08_11	23	192	21	22	3	3	**4**	27	3600
				25,26	0	0	**4**	309	3600
graph_D08_12	23	199	21	22	6	6	**7**	97	3600
				28,29	0	0	**5**	156	3600
graph_D08_13	23	205	21	22	8	8	**8**	29	3600
				30	0	0	**8**	945	3600
graph_D08_14	23	205	21	22	7	7	**7**	25	3600
				29	**0**	**6**	**7**	3600	3600
graph_D08_15	27	270	24	25	6	6	**8**	223	3600
				31,33	0	0	**8**	922	3600
graph_D08_16	27	277	23	24	9	9	**10**	371	3600
				33,34	0	0	**10**	615	3600
graph_D08_17	27	285	25	26	8	8	**10**	189	3600
				34,36	0	0	**10**	1514	3600
graph_D08_18	31	369	28	29	11	11	**15**	551	3600
				40,44	**0**	**11**	**12**	3600	3600

good feasible solutions. For the instances that (IP2-B) could not solve within the time limit, especially for the ones that do not appear in Table 6, we find that the most (or all) of the time limit has been spent in solving the LP relaxation due to our aggressive approach on the blossom cut generation.

4 | CONCLUDING REMARKS

In this paper, we present exact solution algorithms via IP techniques to solve the problem of finding the minimum deficiency for general graphs. Starting with a natural IP formulation of the problem, we devise new formulations with tighter LP relaxations. We present a cutting plane algorithm and an alternative column generation algorithm to solve the LP relaxation of the resulting model. In order to solve the problem to optimality, we incorporate the cutting plane algorithm into a branch-and-bound tree and obtain a branch-and-cut algorithm. Our computational results on a set of random instances indicate that our branch-and-cut algorithm can solve the instances of medium size efficiently. In particular, we find that the algorithm overcomes the two major drawbacks of the existing methods from the literature, namely finding good feasible solutions and providing strong relaxation bounds, thus proving the optimality of the solutions quickly. These improvements can be especially useful for testing the existing (or new) conjectures about the deficiency or the span of the edge-colorings with minimum deficiency, hence contributing to theoretical research in graph theory as well.

We perform the majority of our experiments using $K = \Delta + 1$ in order to work with instances having larger deficiencies. We find that almost all of the instances become interval colorable after the addition of a few more colors. Therefore, when K is large, we propose a framework where we start with $K = \Delta + 1$ and gradually increase its value. This enables us to re-use some information from previous solves such as cutting planes and feasible solutions as warm starts.

In our numerical experiments, we observe that the cutting plane algorithm is comparable with the column generation algorithm in solving the LP relaxations. As such, we do not incorporate the column generation into our final algorithm. However, we also recognize that the column generation might be more efficient for larger and denser instances, which merits further research. On the other hand, although our IP formulation significantly outperforms the CP formulation, a comparison with an improved CP model is also a subject of future research.

ACKNOWLEDGMENTS

We are grateful to Tinaz Ekim and Z. Caner Taskin for providing helpful discussions. The research of Luedtke has been supported in part by the National Science Foundation under grant SES-1422768 and by the Office of Naval Research under award N00014-15-1-2268.

ORCID

Merve Bodur http://orcid.org/0000-0002-9276-3755

REFERENCES

[1] S. Altinakar, G. Caporossi, and A. Hertz, *A comparison of integer and constraint programming models for the deficiency problem*, Comput. Oper. Res. **68** (2016), 89–96.

[2] S. Altinakar, G. Caporossi, and A. Hertz, *Symmetry breaking constraints for the minimum deficiency problem*, J. Graph Algorithms Appl. **21** (2017), 195–218.

[3] A.S. Asratian, C.J. Casselgren, J. Vandenbussche, and D.B. West, *Proper path-factors and interval edge-coloring of (3, 4)-biregular bigraphs*, J. Graph Theory **61** (2009), 88–97.

[4] A. S. Asratian, T.M. Denley, and R. Häggkvist, *Bipartite Graphs and Their Applications*, vol. **131**, Cambridge University Press, Cambridge, 1998.

[5] A.S. Asratian and R.R. Kamalian, *Interval colorings of the edges of a multigraph*, Appl. Math. **5** (1987), 25–34. (in Russian).

[6] A.S. Asratian and R.R. Kamalian, *Investigation on interval edge-colorings of graphs*, J. Combin. Theory, Ser. B **62** (1994), 34–43.

[7] C. Barnhart, E.L. Johnson, G.L. Nemhauser, M.W. Savelsbergh, and P.H. Vance, *Branch-and-price: Column generation for solving huge integer programs*, Oper. Res. **46** (1998), 316–329.

[8] M. Bouchard, A. Hertz, and G. Desaulniers, *Lower bounds and a tabu search algorithm for the minimum deficiency problem*, J. Combin. Optim. **17** (2009), 168–191.

[9] G. Desaulniers, J. Desrosiers, and M.M. Solomon, *Column Generation*, vol. **5**. Springer, Berlin, 2006.

[10] B. Dezső, A. Jüttner, and P. Kovács, *LEMON-an open source C++ graph template library*, Electron. Notes Theor. Comput. Sci. **264** (2011), 23–45.

[11] J. Edmonds, *Maximum matching and a polyhedron with 0, l-vertices*, J. Res. Nat. Bur. Standards B **69** (1965), 125–130.

[12] Y. Feng and Q. Huang, *Consecutive edge-coloring of the generalized* θ*-graph*, Discrete Appl. Math. **155** (2007), 2321–2327.

[13] K. Giaro, *The complexity of consecutive delta-coloring of bipartite graphs: 4 is easy, 5 is hard*, Ars Combin. **47** (1997), 287–298.

[14] K. Giaro and M. Kubale, *Consecutive edge-colorings of complete and incomplete Cartesian products of graphs*, Congr. Numer. **128** (1997), 143–150.

[15] K. Giaro and M. Kubale, *Compact scheduling of zero-one time operations in multi-stage systems*, Discrete Appl. Math. **145** (2004), 95–103.

[16] K. Giaro, M. Kubale, and M. Malafiejski, *Compact scheduling in open shop with zero-one time operations*, INFOR **37** (1999), 37–47.

[17] K. Giaro, M. Kubale, and M. Małafiejski, *On the deficiency of bipartite graphs*, Discrete Appl. Math. **94** (1999), 193–203.

[18] K. Giaro, M. Kubale, and M. Małafiejski, *Consecutive colorings of the edges of general graphs*, Discrete Math. **236** (2001), 131–143.

[19] M. Grötschel and O. Holland, *A cutting plane algorithm for minimum perfect 2-matchings*, Computing **39** (1987), 327–344.

[20] S. Gualandi and F. Malucelli, *Exact solution of graph coloring problems via constraint programming and column generation*, INFORMS J. Comput. **24** (2012), 81–100.

[21] A. Hagberg, D. Schult, and P. Swart, *Exploring network structure, dynamics, and function using networkX*, Proceedings of the 7th Python in Science Conference (Pasadena, CA), G. Varoquaux, T. Vaught, and J. Millman (eds.), (2008), pp. 11–15.

[22] D. Hanson, C. Loten, and B. Toft, *On interval colourings of bi-regular bipartite graphs*, Ars Combin. **50** (1998), 23–32.

[23] R. Kamalian., *Interval edge-colorings of graphs*, Ph.D. thesis, The Institute of Mathematics of the Siberian Branch of the Academy of Sciences of USSR, Novosibirsk, 1990.

[24] R. Kamalian and A. Mirumian, *Interval edge colorings of bipartite graphs of some class*, Dokl NAN RA **97** (1997), 3–5. (in Russian).

[25] R.R. Kamalian, Interval colorings of complete bipartite graphs and trees, preprint, Comp. Cen. Acad. Sci. Armenian SSR, Erevan (1989) (in Russian).

[26] R.R. Kamalian, *On cyclically-interval edge colorings of trees*, Bul. Acad. Stiinte Repub. Mold. Mat. **68** (2012), 50–58.

[27] R.R. Kamalian, On one-sided interval edge colorings of biregular bipartite graphs. 2012. http://arxiv.org/abs/1205.0130.

[28] R.R. Kamalian, *On a number of colors in cyclically interval edge colorings of simple cycles*, Open J. Discrete Math. **3** (2013), 43.

[29] D. König, *Über graphen und ihre anwendung auf determinantentheorie und mengenlehre*, Math. Ann. **77** (1916), 453–465.

[30] M. Kubale, *Graph Colorings*, vol. **352**, American Mathematical Society, Providence, RI, 2004.

[31] J. Lee and J. Leung, *A comparison of two edge-coloring formulations*, Oper. Res. Lett. **13** (1993), 215–223.

[32] A.N. Letchford, G. Reinelt, and D.O. Theis, "A faster exact separation algorithm for blossom inequalities," *International Conference on Integer Programming and Combinatorial Optimization*, Springer (2004), pp. 196–205.

[33] F. Margot, "Symmetry in integer linear programming," *50 Years of Integer Programming 1958–2008*, Springer, Berlin (2010).

[34] G.L. Nemhauser and S. Park, *A polyhedral approach to edge coloring*, Oper. Res. Lett. **10** (1991), 315–322.

[35] M.W. Padberg and M.R. Rao, *Odd minimum cut-sets and b-matchings*, Math. Oper. Res. **7** (1982), 67–80.

[36] P.A. Petrosyan, *Interval edge colorings of mobius ladders*. Proceedings of the CSIT Conference, (2007), pp. 146–149 (in Russian).

[37] P.A. Petrosyan, *Interval edge-colorings of complete graphs and n-dimensional cubes*, Discrete Math. **310** (2010), 1580–1587.

[38] P.A. Petrosyan, On interval non-edge-colorable Eulerian multigraphs, 2013, http://arxiv.org/abs/1311.2210.

[39] P.A. Petrosyan, H. Arakelyan, and V. Baghdasaryan, *A generalization of interval edge-colorings of graphs*, Discrete Appl. Math. **158** (2010), 1827–1837.

[40] A. Pyatkin, *Interval coloring of (3, 4)-biregular bipartite graphs having large cubic subgraphs*, J. Graph Theory **47** (2004), 122–128.

[41] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, vol. **24**. Springer, Berlin, 2003.

[42] A. Schwartz, *The deficiency of a regular graph*, Discrete Math. **306** (2006), 1947–1954.

[43] S. Sevastjanov, *Interval colorability of the edges of a bipartite graph*, Met. Diskret. Analiz **50** (1990), 61–72. (in Russian).

[44] P.G. Tait, *Note on the coloring of maps*, Proc. Royal Soc. Edinburgh **10** (1880), 727–728.

[45] V. Vizing, *On an estimate of the chromatic class of a p-graph*, Diskret. Analiz **3** (1964), 25–30.

[46] F. Yang and X. Li, *Interval coloring of (3,4)-biregular bigraphs having two (2,3)-biregular bipartite subgraphs*, Appl. Math. Lett. **24** (2011), 1574–1577.

How to cite this article: Bodur M, Luedtke JR. Integer programming formulations for minimum deficiency interval coloring. *Networks.* 2018;72:249–271. https://doi.org/10.1002/net.21826

APPENDIX: PROOFS

Proposition 1. *(IP2) is a formulation of (MinDef) problem.*

Proof. Let $c : E \to \mathcal{K}$ be a proper edge-coloring of graph $G = (V, E)$, where $\mathcal{K} = \{0, 1, \ldots, K-1\}$. First, we need to show that there is a feasible solution of (IP2) whose objective function value is less than or equal to the deficiency of c, which is

$$\mathrm{def}(G, c) = \sum_{i \in V} \left(\max_{j \in N(i)} c(i,j) - \min_{j \in N(i)} c(i,j) - d(i) + 1 \right).$$

For each $i \in V$, we define

$$\ell^*(i) = \min_{j \in N(i)} c(i,j) \quad \text{and} \quad u^*(i) = \max_{j \in N(i)} c(i,j).$$

Also, we let

$$x^*_{ijk} = \begin{cases} 1, & \text{if } c(i,j) = k \\ 0, & \text{o.w.} \end{cases}, \quad \{i,j\} \in E,\ k \in \mathcal{K},$$

and

$$y^*_{i,[\ell,u]} = \begin{cases} 1, & \text{if } \ell = \ell^*(i) \quad \text{and } u = u^*(i) \\ 0, & \text{o.w.} \end{cases}, \quad i \in V,\ [\ell,u] \in \mathcal{I}^i.$$

At (x^*, y^*), all constraints in (IP2), except (2c), are trivially satisfied by the definitions of x^* and y^*. Now, we confirm that (2c) also holds at (x^*, y^*). Let $i \in V$ and $k \in \mathcal{K}$. Then, $\sum_{j \in N(i)} x^*_{ijk} = 1$ if $c(i,j) = k$ for some $j \in N(i)$. In that case, we have $\ell^*(i) \le k \le u^*(i)$, which means that $[\ell^*(i), u^*(i)] \in \mathcal{I}^i(k)$. Hence, $\sum_{[\ell,u] \in \mathcal{I}^i(k)} y^*_{i,[\ell,u]} = 1$. Otherwise, as $\sum_{j \in N(i)} x^*_{ijk} = 0$, the constraint is also satisfied. Therefore, (x^*, y^*) is a feasible solution to (IP2), whose objective value is equal to

$$\sum_{i \in V} \sum_{[\ell,u] \in \mathcal{I}^i} (u - \ell - d(i) + 1)\, y^*_{i,[\ell,u]} = \sum_{i \in V} (u^*(i) - \ell^*(i) - d(i) + 1) = \operatorname{def}(G, c).$$

For the reverse direction, we need to show that any feasible solution of (IP2) corresponds to a proper edge-coloring of G with deficiency less than or equal to the objective function value. Let $(\hat{x}, \hat{y})$ be a feasible solution of (IP2). We define $\hat{c} : E \to \mathcal{K}$ as

$$\hat{c}(i,j) = \sum_{k \in \mathcal{K}} k\, \hat{x}_{ijk}, \quad \{i,j\} \in E.$$

From (2a), we guarantee that each edge takes exactly one color from $\mathcal{K}$ in $\hat{c}$. For $i \in V$ and $k \in \mathcal{K}$, as we have

$$\sum_{j \in N(i)} \hat{x}_{ijk} \overset{(2c)}{\le} \sum_{[\ell,u] \in \mathcal{I}^i(k)} \hat{y}_{i,[\ell,u]} \le \sum_{[\ell,u] \in \mathcal{I}^i} \hat{y}_{i,[\ell,u]} \overset{(2b)}{=} 1,$$

$\hat{c}$ is a proper edge-coloring. Next, for each $i \in V$, let $\hat{\ell}(i), \hat{u}(i) \in \mathcal{K}$ such that $\hat{y}_{i,[\hat{\ell}(i),\hat{u}(i)]} = 1$. Then, the objective function value at $(\hat{x}, \hat{y})$ is $\sum_{i \in V} (\hat{u}(i) - \hat{\ell}(i) - d(i) + 1)$. We know that $\hat{\ell}(i) \le \hat{c}(i,j) \le \hat{u}(i)$ for all $i \in V,\ j \in N(i)$. Therefore, as $\max_{j \in N(i)} \hat{c}(i,j) \le \hat{u}(i)$ and $\min_{j \in N(i)} \hat{c}(i,j) \ge \hat{\ell}(i)$, $\operatorname{def}(G, \hat{c})$ is smaller than or equal to the objective value of $(\hat{x}, \hat{y})$, which completes the proof. ■

Proposition 2. *(IP3) is a formulation of (MinDef) problem.*

Proof. Let $c : E \to \mathcal{K}$ be a proper edge-coloring of graph $G = (V, E)$, where $\mathcal{K} = \{0, 1, \ldots, K-1\}$. First, we need to show that there is a feasible solution of (IP3) whose objective function value is less than or equal to the deficiency of c, which is

$$\operatorname{def}(G, c) = \sum_{i \in V} \left(\max_{j \in N(i)} c(i,j) - \min_{j \in N(i)} c(i,j) - d(i) + 1 \right).$$

For each $i \in V$, let

$$u^*_i = \max_{j \in N(i)} c(i,j) \quad \text{and} \quad \ell^*_i = \min_{j \in N(i)} c(i,j)$$

and define

$$y^*_{i,[\ell,u]} = \begin{cases} 1, & \text{if } \ell = \ell^*_i \text{ and } u = u^*_i \\ 0, & \text{o.w.} \end{cases},$$

then it is easy to see that (7a) and (7f) are satisfied. Moreover, the objective function value of (IP3) is equal to $\mathrm{def}(G,c)$. Now, for each $k \in \mathcal{K}$, let $E_k = \{e \in E : c(e) = k\}$. Note that E_k might be empty for some $k \in \mathcal{K}$. Then, $E_0, \ldots, E_{K-1}$ is a partition of E, where E_k is a matching for each $k \in \mathcal{K}$. If we define

$$z^*_{k,m} = \begin{cases} 1, & \text{if } m = E_k \text{ and } E_k \neq \emptyset \\ 0, & \text{o.w.} \end{cases}, \ k \in \mathcal{K},\ m \in \mathcal{M},$$

then (7e) is satisfied. Next, we check the remaining constraints to show that (x^*, y^*) is feasible to (IP3):

$$\text{(7b)} : \text{For } e \in E,\ \sum_{m\in\mathcal{M}}\sum_{k\in\mathcal{K}} a_{m,e} z^*_{k,m} = \sum_{k\in\mathcal{K}} b_{E_k,e} z^*_{k,E_k} = z^*_{c(e),E_{c(e)}} = 1.$$

$$\text{(7c)} : \text{For } k \in \mathcal{K},\ \sum_{m\in\mathcal{M}} z^*_{k,m} = z^*_{k,E_k} \leq 1.$$

$$\text{(7d)} : \text{For } i \in V,\ k \in \mathcal{K},\ \sum_{m\in\mathcal{M}} b_{m,i} z^*_{k,m} = b_{E_k,i} z^*_{k,E_k} = \begin{cases} 1, & \text{if } \exists\ \{i,j\} \in E \text{ s.t. c(i, j) = k} \\ 0, & \text{o.w.} \end{cases}.$$

The constraint is satisfied in both cases as

$$b_{E_k,i} z^*_{k,E_k} = 1 \Rightarrow \ell^*_i \leq k \leq u^*_i \Rightarrow [\ell^*_i, u^*_i] \in \mathcal{I}^i(k) \Rightarrow \sum_{[\ell,u]\in\mathcal{I}^i(k)} y^*_{i,[\ell,u]} = 1.$$

Therefore, (z^*, y^*) is a feasible solution of (IP3) with objective function value $\mathrm{def}(G,c)$.

For the reverse direction, we need to show that any feasible solution of (IP3) corresponds to a proper edge-coloring of G with deficiency less than or equal to the objective function value. Let $(\hat{z}, \hat{y})$ be a feasible solution of (IP3). Letting

$$\hat{\delta}_{e,k} := \sum_{m\in\mathcal{M}} a_{m,e}\, \hat{z}_{k,m},\ e \in E,\ k \in \mathcal{K},$$

we define $\hat{c} : E \to \mathcal{K}$ as

$$\hat{c}(e) = \sum_{k\in\mathcal{K}} k\, \hat{\delta}_{e,k},\ e \in E.$$

(7b) implies that for any $e \in E$, we have $\sum_{k\in\mathcal{K}} \hat{\delta}_{e,k} = 1$. Then, as $\hat{\delta}_{e,k}$ values are nonnegative integers, this means that each edge $e \in E$ takes exactly one color, which is $\hat{c}(e)$. This also shows that edge colors are actually taken from the set $\mathcal{K}$. Next, we show that $\hat{c}$ is a proper edge-coloring. Let $i \in V$ and $k \in \mathcal{K}$. Then, we have

$$\sum_{j\in N(i)} \hat{\delta}_{\{i,j\},k} = \sum_{e=\{i,j\}:j\in N(i)} \sum_{m\in\mathcal{M}} a_{m,e}\, \hat{z}_{k,m} = \sum_{m\in\mathcal{M}} \hat{z}_{k,m} b_{m,i} \overset{(7d)}{\leq} \sum_{[\ell,u]\in\mathcal{I}^i(k)} y^*_{i,[\ell,u]} \overset{(7f)}{\leq} \sum_{[\ell,u]\in\mathcal{I}^i} y^*_{i,[\ell,u]} \overset{(7a)}{=} 1,$$

which means that each color can be used at most once for the set of edges incident to a vertex. Finally, letting

$$\hat{u}_i = \max_{j\in N(i)} \hat{c}(i,j) \quad \text{and} \quad \hat{\ell}_i = \min_{j\in N(i)} \hat{c}(i,j)$$

for all $i \in V$, we consider the deficiency of $\hat{c}$:

$$\mathrm{def}(G, \hat{c}) = \sum_{i \in V} (\hat{u}_i - \hat{\ell}_i - d(i) + 1).$$

For any $i \in V$, (7d) for $k = \hat{u}_i$ and for $k = \hat{\ell}_i$ imply that $\hat{y}_{i,[\ell,u]} = 1$ for some $[\ell, u] \in \mathcal{I}^i$ with $\ell \leq \hat{u}_i \leq u$ and $\ell \leq \hat{\ell}_i \leq u$, respectively. This combined with (7a) shows that $\hat{y}_{i,[\ell,u]} = 1$ for the interval $[\ell, u] \in \mathcal{I}^i$ with $u \geq \hat{u}_i$ and $\ell \leq \hat{\ell}_i$. Hence, the objective function value of the solution $(\hat{x}, \hat{y})$ is greater than or equal to $\mathrm{def}(G, \hat{c})$. ∎

Proposition 3. *Assume that $K \geq \Delta + 1$ and $\Delta > 0$. Let $z^* \in \mathbb{R}_+^{|\mathcal{M}|}$ and v^* be an optimal solution and the optimal value of (EC), respectively. Then, $(\hat{z}, \hat{y})$ with*

$$\hat{z}_{k,m} = \begin{cases} z_m^*/v^*, & \text{if } k < \Delta \\ (v^* - \Delta) z_m^*/v^*, & \text{if } k = \Delta, \\ 0, & \text{o.w.} \end{cases} \quad k \in \mathcal{K},\ m \in \mathcal{M}$$

and

$$\hat{y}_{i,[\ell,u]} = \begin{cases} 1, & \text{if } \ell = 0 \text{ and } u = K-1 \\ 0, & \text{o.w.} \end{cases}, \quad i \in V,\ [\ell, u] \in \mathcal{I}^i$$

is a feasible solution to (LP3).

Proof. Vizing's Theorem [45] implies that $v^* \leq \Delta + 1$. Also, we have $v^* = \sum_{m \in \mathcal{M}} z_m^*$. Therefore, the bound constraints of (LP3) are satisfied by $(\hat{z}, \hat{y})$. Now, we show that the rest of the constraints in (LP3) also hold at $(\hat{z}, \hat{y})$:

(7a): For any $i \in V$, only $\hat{y}_{i,[0,K-1]} = 1$. Therefore, $\sum_{[\ell,u] \in \mathcal{I}^i} \hat{y}_{i,[\ell,u]} = 1,\ i \in V$.

(7b): Let $e \in E$. As we have, for each $m \in \mathcal{M}$,

$$\sum_{k \in \mathcal{K}} \hat{z}_{k,m} = \sum_{k < \Delta} \hat{z}_{k,m} + \hat{z}_{\Delta,m} + \sum_{k > \Delta} \hat{z}_{k,m} = \Delta z_m^*/v^* + (v^* - \Delta) z_m^*/v^* = z_m^*,$$

we obtain

$$\sum_{m \in \mathcal{M}} \sum_{k \in \mathcal{K}} a_{m,e}\, \hat{z}_{k,m} = \sum_{m \in \mathcal{M}} a_{m,e} \sum_{k \in \mathcal{K}} \hat{z}_{k,m} = \sum_{m \in \mathcal{M}} a_{m,e}\, z_m^* = 1.$$

(7c): For any $k < \Delta$,

$$\sum_{m \in \mathcal{M}} \hat{z}_{k,m} = \frac{1}{v^*} \sum_{m \in \mathcal{M}} z_m^* = \frac{1}{v^*} v^* = 1.$$

For $k = \Delta$,

$$\sum_{m \in \mathcal{M}} \hat{z}_{\Delta,m} = \frac{v^* - \Delta}{v^*} \sum_{m \in \mathcal{M}} z_m^* = \frac{v^* - \Delta}{v^*} v^* = v^* - \Delta \leq \Delta + 1 - \Delta = 1.$$

For $k > \Delta$, we have $\hat{z}_{k,m} = 0,\ m \in \mathcal{M}$ so the rest of this type of constraints is also satisfied.

(7d): For any $i \in V, k \in \mathcal{K}$,

$$\sum_{m \in \mathcal{M}} b_{m,i} \hat{z}_{k,m} \leq \sum_{m \in \mathcal{M}} \hat{z}_{k,m} = \begin{cases} \sum_{m \in \mathcal{M}} z_m^*/v^* = 1, & \text{if } k < \Delta \\ (v^* - \Delta), & \text{if } k = \Delta \\ 0, & \text{o.w.} \end{cases} \leq 1 = \sum_{[\ell,u] \in \mathcal{I}^i(k)} \hat{y}_{i,[\ell,u]}.$$

∎