Fractional Chromatic Numbers from Decision Diagrams

Timo Brand[1] and Stephan Held[2]

[1]University of Bonn,Germany, Timo.Brand@uni-bonn.de
[2]Research Institute for Discrete Mathematics and Hausdorff Center for Mathematics, University of Bonn, Germany, held@dm.uni-bonn.de

March 7, 2024

Abstract

Recently, Van Hoeve proposed an algorithm for graph coloring based on an integer flow formulation on decision diagrams for stable sets [7]. In this paper, we prove that the solution to the linear relaxation of an exact decision diagram determines the fractional chromatic number of a graph.

We also conduct experiments using exact decision diagrams and could determine the chromatic number of `r1000.1c` from the DIMACS benchmark set. It was previously unknown and is the first newly solved DIMACS instance in over 10 years.

Keywords: Graph coloring, Decision diagrams, Integer programming

1 Introduction

A (vertex) coloring of a graph $G = (V, E)$ assigns a color to each vertex such that adjacent vertices have different colors. Thus, each set of vertices with the same color is a *stable set* (also called *independent set*) in G. The *(vertex) coloring problem* is to compute a coloring with the minimum possible number of colors. This number is denoted by $\chi(G)$ and also called the *chromatic number* of G.

Let $\mathcal{S}$ denote the set of all *stable sets* in G. Then, solving the following integer programming model yields the chromatic number [6]:

$$\begin{aligned}\chi(G) = \min &\sum_{S\in\mathcal{S}} z_S \\ \text{s.t.} &\sum_{S\in\mathcal{S}:j\in S} z_i \geq 1 \quad \forall j \in V \\ &z_S \in \{0,1\} \quad \forall S \in \mathcal{S}\end{aligned} \tag{VCIP}$$

It is a special case of the *set cover problem*, where the vertex set V has to be covered with a minimum number of stable sets. The linear relaxation results in the *fractional chromatic number* $\chi_f(G)$.

$$\begin{aligned}\chi_f(G) := \min &\sum_{S\in\mathcal{S}} z_I \\ \text{s.t.} &\sum_{S\in\mathcal{S}:j\in S} z_S \geq 1 \quad \forall j \in V \\ &0 \leq z_S \leq 1 \quad \forall S \in \mathcal{S}.\end{aligned} \tag{VCLP}$$

Lovász showed that $\chi(G) \leq \mathcal{O}(\log n) \cdot \chi_f(G)$ [4]. However for any $\epsilon > 0$, approximating either $\chi_f(G)$ or $\chi(G)$ within a factor $1^{1-\epsilon}$ is NP-hard [8]. The IP formulation (VCIP) is the basis for several branch-&-price algorithms for vertex coloring [6, 5, 3].

Recently, Van Hoeve [7] proposed to tackle the graph coloring problem with decision diagrams. A (binary) decision diagram consists of an acyclic digraph with arc labels. It can represent the set of feasible solutions to an optimization problem P, e.g. the stable sets of a graph. They are called exact if they represent all solutions, e.g. all stable sets. Van Hoeve showed how to compute a graph coloring using a constrained network flow in the decision diagram.

1.1 Contributions

We show that the linear relaxation of the integral network flow in an exact decision diagram describes the fractional chromatic number, i.e. the linear relaxations arising from (VCLP) and from exact decision diagrams lead to the same lower bound. Thus, fractional flows in relaxed decision diagrams provide fast lower bounds for the (fractional) chromatic number.

Finally, we show that the exact decision diagrams are computationally efficient on dense instances. For the first time, we can compute the chromatic number of `r1000.1c` and improve the best known lower bound of `DJSC500.9`.

The paper is organized as follows. In Section 2, we shortly describe decision diagrams for the stable set problem and how a graph coloring integer program based on such a decision diagram can be formulated (Section 2.3) as proposed by [7]. Then, in Section 3 we prove that the solution to the linear relaxation of the integer program determines the fractional chromatic number. Section 4 contains experimental results on the newly solved DIMACS instance `r1000.1c`, followed by Conclusions.

2 Decision Diagrams

Here, recap how stable sets can be represented through decision diagrams as proposed in [1, 7].

2.1 Decision Diagrams in General

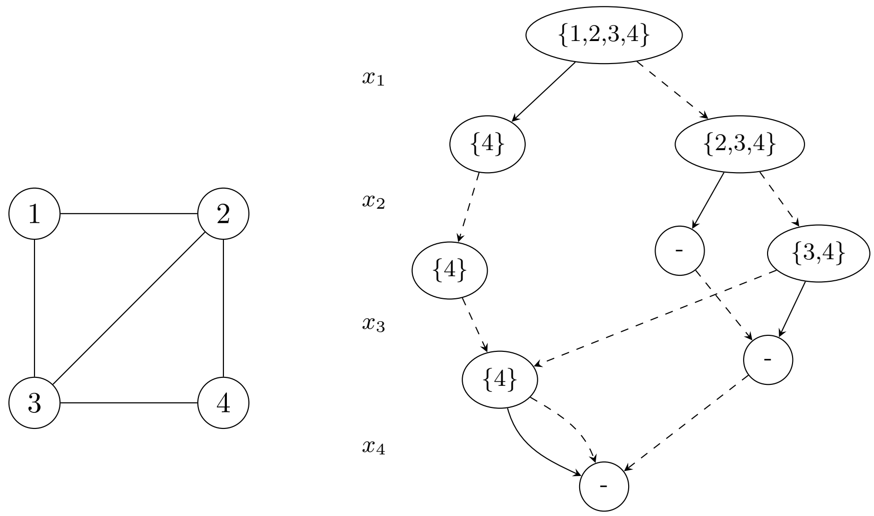
Consider the decision variables $X = \{x_1, x_2, \ldots, x_n\} \in \{0,1\}^n$ of an optimization problem P. Decision diagrams are a possible way to represent the solution space $\mathrm{Sol}(P)$.

A *decision diagram* consists of a layered directed acyclic graph $D = (N, A)$. D has $n+1$ layers $L_1, \ldots, L_{n+1}$, where $N = L_1 \dot\cup \ldots \dot\cup L_{n+1}$. Arcs go only from nodes in one layer to nodes in the next layer. The first layer L_1 consists of a single *root node* r. Likewise, the last layer L_{n+1} consists only of a single *terminal node* t. A layer L_j $(1 \leq j \leq n)$ is a collection of nodes of D. Layer j is associated with the decision variable $x_j \in X$. For a node $u \in L_j$ we denote its layer j by $L(u)$.

Furthermore, the decision diagram has arc labels $l : A \to \{0,1\}$. Arcs are called 0-arcs or 1-arcs depending on their labels. A label encodes if the decision variable of the head level is set to 0 or 1, as we will describe later. For each node, except the terminal node, we have exactly one outgoing 0-arc and at most one outgoing 1-arc.

Each node and each arc must belong to a path from r to t. Given the arcs $(a_1, a_2, \ldots, a_n)$ of an r-t path, we can define a variable assignment of X by setting $x_j = l(a_j)$ for $j = 1, \ldots, n$. $\mathrm{Sol}(D)$ denotes the collection of variable assignments obtained by all r-t paths. A decision diagram D for problem P is called *exact* if $\mathrm{Sol}(P) = \mathrm{Sol}(D)$ and *relaxed* if $\mathrm{Sol}(P) \subseteq \mathrm{Sol}(D)$.

Systematic ways to compute such decision diagrams can be found in [1].



(a) Example graph

(b) Exact decision diagram

Figure 1: An example graph and its exact decision diagram.

2.2 Decision Diagrams for Stable Sets

Let $G = (V, E)$ be a graph with vertex set $V = \{v_1, \dots, v_n\}$. To encode the stable set problem, X contains a decision variable for each vertex in V. Now, an (exact) decision diagram D with labels l for the stable set problem on G consists of $n+1$ layers. For each stable set S in G, there is exactly one path $(a_1, \dots, a_n)$ in D with $l(a_i) = \mathbb{1}_S(v_i)$ and vice versa, where $\mathbb{1}_S$ is the incidence vector of S.

Figure 1 shows a graph G on 4 vertices and an exact decision diagram with 5 layers representing all stable sets. The top layer contains the root r and the bottom layer the terminal t. 1-arcs are drawn as solid lines and 0-arcs as dashed lines.

Notice how the 1-arcs on a path from the root to the terminal correspond to the vertices of a stable set in the graph. Likewise for each stable set in G we can find a corresponding path in D. Thus, $\mathrm{Sol}(P) = \mathrm{Sol}(D)$.

The node labels in the decision diagram show the set of *eligible vertices* at each node $v \in N$, i.e. the vertices that can be still be added individually to the stable sets corresponding to the r-v sub-paths.

In the stable set case, nodes in a common layer with the same set of eligible vertices are called *equivalent*. For general decision diagrams, two nodes v, v' are equivalent if the two subgraphs induced by all v-t-paths and all v'-t-paths are isomorphic. For the stable set problem both notions are equivalent. Equivalent nodes can be contracted into a single node on that layer. This is used to reduce the size of a decision diagram for the stable set problem. Decision diagrams without equivalent nodes are called *reduced decision diagrams*.

Bergman et al. ([1], Algorithm 1) proposed a top-down compilation technique to compute the exact reduced decision diagram D for stable sets.

Decision diagrams can have an exponential size. They can be computed efficiently if the number of nodes per layer is small after contracting equivalent nodes, which is often the case for dense graphs.

2.3 Graph Coloring from Stable Set Decision Diagrams

For graph coloring, Van Hoeve [7] proposed the following constrained integral network flow problem (F) on a stable set decision diagram $D = (N, A)$:

$$
\begin{aligned}
(F) = \ \min \ & \sum_{a \in \delta^+(r)} y_a && && (1)\\
\text{s.t.} \ & \sum_{a=(u,v) \mid L(u)=j, \ell(a)=1} y_a \geq 1 && \forall j \in V && (2)\\
& \sum_{a \in \delta^-(u)} y_a - \sum_{a \in \delta^+(u)} y_a = 0 && \forall u \in N \setminus \{r, t\} && (3)\\
& y_a \in \{0, \dots, n\} && \forall a \in A && (4)
\end{aligned}
$$

The constraints (1)–(4) encode an integral r-t-flow that implicitly covers each original vertex through an activating arc (2). The flow can be decomposed into paths, which correspond to stable sets. Minimizing the flow value corresponds to minimize the number of paths and, thus, to minimizing the number of stable sets in a stable set cover.

Van Hoeve [7, Theorem 2] shows that (1)–(4) computes the chromatic number, if D is an exact decision diagram. [1]

Relaxed decision diagrams might additionally contain paths $(a_1, \dots, a_n)$ that do not represent stable sets.

The main emphasis of Van Hoeve's work [7] is an iterative method to compute the chromatic number based on relaxed decision diagrams. While exact decision diagrams can have an exponential size, he begins with a relaxed decision diagram that approximates $\mathrm{Sol}(P)$. Solving the network flow

[1]Formally, Van Hoeve proved this for the partitioning formulation of (F), where (2) are equality constraints. Both formulations are equivalent and he also uses the covering formulation in his implementation.

problem (F) on this relaxation yields a lower bound to the chromatic number and enough information to refine the relaxed decision diagram to become a better approximation. The refinement continuous until an optimum coloring is found (or a time limit is reached). He also provides a class of instances, where the refinement approach ends with a polynomial size decision diagram, where an exact decision diagram has exponential size [7, Theorem 7]. Van Hoeve was able to report a new lower bound of 145 for instance `C2000.9`.

3 The Fractional Chromatic Number and Decision Diagrams

In this section we prove our main observation. For exact decision diagrams, the fractional chromatic number $\chi_f(G)$ is determined by the linear relaxation (F') of (F), where the linear relaxation (F') is defined as

$$
\begin{aligned}
(F') = \ \min \ & \sum_{a \in \delta^+(r)} y_a && && (5)\\
\text{s.t.} \ & \sum_{a=(u,v) \mid L(u)=j, \ell(a)=1} y_a \geq 1 && \forall j \in V && (6)\\
& \sum_{a \in \delta^-(u)} y_a - \sum_{a \in \delta^+(u)} y_a = 0 && \forall u \in N \setminus \{r, t\} && (7)\\
& 0 \leq y_a \leq n && \forall a \in A && (8)
\end{aligned}
$$

We have the following result.

Theorem 1. *Let $G = (V, E)$ be a graph and $D = (N, A)$ an exact stable set decision diagram for G. Then the linear relaxation (F') is equivalent to (VCLP).*

Proof. We show how optimum solutions can be transformed between the two problem formulations.

Let $(z_S)_{S \in \mathcal{S}}$ be an optimum basic solution of (VCLP). We transform into a solution $y_a, a \in A$ of (F') with the same objective value..

Foreach $S \in \mathcal{S}$ with $z_S > 0$, choose a path $(a_1, \ldots, a_n)$ in the decision diagram, where $l(a_j) = 1$ if $j \in S$ and $l(a_j) = 0$ if $(j \notin S)$. Since S is a stable set and D is exact, such a path must exist. Increase the flow along this path by z_i.

As z is an optimum basic solution, it uses at most n stable sets with positive value. Since $0 \leq z \leq 1$, constraints (8) are satisfied. Since we augment the flow along s-t-paths, the flow condition (7) is also fulfilled. For each vertex $j \in V$ we have $\sum_{S \in \mathcal{S}: j \in S} z_S \geq 1$. This implies that at least on unit of flow is sent through 1-arcs in layer j of the decision diagram, thus the solution satisfies (6) as well and is a valid solution of (F'). By construction the objective values of both solutions coincide.

For the other direction, let $(y_a)_{a \in A}$ be an optimum fractional solution of (F') . It is an r-t network flow that can be decomposed into flows along r-t-paths, where no path is repeated. Let $P_1, \ldots, P_k$ be such a decomposition. Each path P_i $(i \in [k])$ corresponds to a unique stable set $S_i \in \mathcal{S}$ since D is an exact decision diagram. We construct a solution z of (VCLP) by setting z_{S_i} to the of flow sent along path P_i for $in \in [k]$ and $z_S = 0$ for all stable sets that are not represented in the decomposition. By optimality, $z \leq 1$ and the variable bounds are satisfied. By (6), the amount of flow on 1-arcs in layer j is at least 1. This implies that for each vertex $j \in V$, $\sum_{S \in \mathcal{S}: j \in S} z_S \geq 1$, and z is a feasible solution to (VCLP) with the same objective value as y.

We conclude that for an exact decision diagram of a graph G, (F') computes the fractional chromatic number $\chi_f(G)$. □

From Theorem 1, we can also conclude that the linear program (F') always has an optimum solution with a polynomial-size support, given an exact decision diagram.

Corollary 1. *Let $G = (V, E)$ be a graph and $D = (N, A)$ an exact stable set decision diagram for G, there is an optimum solution y to (F') with $|\{a \in A, y_a > 0\}| \leq n^2$, where $n = |V|$.*

Proof. The transformation of a basic optimum solution of (VCLP) results in an optimum solution y that consists of at most n paths, where each path has at most n edges. □

As the underlying decision diagram and, thus, the support of a basic optimum solution to (F') can have exponential size, the transformation of a solution y for (F') might yield an optimum solution z for (VCLP) that is not a basic solution.

A nice property of (F') is that the formulation provides a lower bound on $\chi_f(G)$ for each relaxed decision diagram. Good bounds might be easier to compute in practice than for (VCLP), where the branch-&-price algorithm can take many iterations with weigthed stable set problems in the pricing problem.

Lovász [4] showed that the ratio between $\chi(G)$ and $\chi_f(G)$ is in $\mathcal{O}(\log n)$. With this, we obtain the following corollary:

Corollary 2. *Given an exact decision diagram, the integrality gap between* (F) *and* (F') *is* $\mathcal{O}(log\ n)$.

4 Newly solved DIMACS instance `r1000.1c`

We re-implemented and essentially verified the experimental results of Van Hoeve [7] using relaxed decision diagrams. Van Hoeve also showed for which instances, the chromatic number can be computed efficiently within 1 hour of running time using the exact decision diagram.

Spending a little more time we were able to compute the chomratic number of the DIMACS benchmark instance `r1000.1c`. It is $\chi(r1000.1c) = 98$.

CPLEX solves the integer program in ..., but it might run into numerical errors. Therefore we also solved the integer program uisng SCIP-exact by Eifler and Gleixner [2] to verify $\chi(r1000.1c) = 98$.

5 Conclusions

We showed that the fractional flow formulation for graph coloring applied to exact decision diagrams introduced by Van Hoeve [7], determines the fractional chromatic number of a graph. It is an interesting alternative to using the set cover formulation, as it can be faster to compute on certain instances. The fractional lower bounds from relaxed decision diagrams presented in [7] are thus lower bounds for the fractional chromatic number.

References

[1] D. Bergman, A. A. Cire, W.-J. v. Hoeve, and J. N. Hooker. Optimization bounds from binary decision diagrams. *INFORMS Journal on Computing*, 26(2):253–268, 2014.

[2] L. Eifler and A. Gleixner. A computational status update for exact rational mixed integer programming. *Mathematical Programming*, 197(2):793–812, 2023.

[3] S. Held, W. Cook, and E. C. Sewell. Maximum-weight stable sets and safe lower bounds for graph coloring. *Mathematical Programming Computation*, 4(4):363–381, 2012.

[4] L. Lovász. On the ratio of optimal integral and fractional covers. *Discrete mathematics*, 13(4):383–390, 1975.

[5] E. Malaguti, M. Monaci, and P. Toth. An exact approach for the vertex coloring problem. *Discrete Optimization*, 8(2):174—-190, 2011.

[6] A. Mehrotra and M. A. Trick. A column generation approach for graph coloring. *INFORMS Journal on Computing*, 8(4):344–354, 1996.

[7] W.-J. van Hoeve. Graph coloring with decision diagrams. *Mathematical Programming*, pages 1–44, 2021.

[8] D. Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. *Theory Of Computing*, 3:103–128, 2007.