Journal of Graph Algorithms and Applications
http://jgaa.info/ vol. 28, no. 1, pp. 365–383 (2024)
DOI: 10.7155/jgaa.v28i1.2964

A Greedy Probabilistic Heuristic for Graph Black-and-White Anticoloring

Daniel Berend [1,2]
Shaked Mamana [3]

[1]Department of Mathematics and Computer Science, Ben-Gurion University, Beer Sheva 84105, Israel.
[2]Research supported in part by The Milken Families Foundation Chair in Mathematics.
[3]Department of Mathematics, Ben-Gurion University, Beer Sheva 84105, Israel.

Submitted: June 2023	Accepted: March 2024	Published: August 2024
Article type: Regular paper	Communicated by: Y. Okamoto	

Abstract. Given a graph G and positive integers b and w, the *Black-and-White Coloring problem* asks about the existence of a partial vertex-coloring of G, with b vertices colored black and w white, such that there is no edge between a black and a white vertex. The problem is known to be NP-complete.

In this paper, we deal with the optimization version, mainly for random graphs. Using the method of conditional expectations, we develop a heuristic with a good performance. We also obtain theoretical results on some of the relevant quantities, and compare the performance of the heuristic with that of several others.

1 Introduction

An *anticoloring* of a graph is a coloring of some of the vertices, such that no two adjacent vertices are colored in distinct colors. In the basic anticoloring problem, we are given a graph $G = (V, E)$, where $|V| = n$, and positive integers $b_1, ..., b_k$, and have to determine whether there exists an anticoloring of G such that b_i vertices are colored in color i, $i = 1, ..., k$ (thus leaving $n - \sum_{i=1}^{k} b_i$ vertices uncolored). The anticoloring problem with $k = 2$ is the *Black-and-White Coloring problem* (henceforward BWC).

The problem has been studied mainly in its optimization version, in which we are given a graph G and a positive integer b, and have to color b of the vertices in black, so that there will remain as many vertices as possible which are non-adjacent to any of those b vertices. (These latter vertices are to be colored in white.) We denote by W_b the maximum possible number of such vertices, by

E-mail addresses: berend@math.bgu.ac.il (Daniel Berend) mamanas@post.bgu.ac.il (Shaked Mamana)

B the set of the vertices we color in black, by W the set of all vertices we can color in white after those of B have been colored in black, and by C the set of vertices we leave uncolored.

The BWC problem was introduced by Berge (see (10)): Given positive integers n and b, place b black and w white queens on an $n \times n$ chessboard, so that no black queen and white queen attack each other, with w as large as possible. (This problem itself is not tackled in this paper, and its complexity remains open.) The BWC problem was introduced and proved to be NP-hard by Hansen, Hertz, and Quinodoz (10). In the same paper, they gave an $O(n^3)$ algorithm for trees, that was later superseded by an $O(n^2 \log^3 n)$ algorithm (7). Kobler, Korach, and Hertz (11) gave a polynomial algorithm for partial k-trees with fixed k. In (5), the BWC problem has been studied for planar graphs, and an algorithm with some bound on the additive error relative to the optimal solution was given. In addition, an optimal solution for a special instance of the problem, namely, the problem suggested by Berge, using kings instead of queens (4), and knights instead of queens (13), was given, and the problem was studied on other families of graphs as well (3; 2; 6; 17). Various algorithms have been suggested for the problem (6; 14; 18).

In this paper, we start with a naive approach for solving the problem. We color uniformly random b vertices in black, and then color in white every vertex that is not a neighbor of a black vertex. Analyzing this approach and using conditional expectations, we are led to develop an improved heuristic. The improvement lies in going over the vertices in a "smart" order and not in a random order. In each step, we color in black the vertex v_0, for which the conditional expectation of the size of the white vertices set, given that v_0 color in black, is maximal. Note that, while we try to color in black at each step the most promising vertex, the calculation of the "profit" resulting from each possible choice is based on the assumption that the rest of the process will continue naively according to a naive approach.

For some families of random graphs, our approach yields the optimal anticoloring with high probability. In the general case, we obtain a lower bound on the performance. We also compare its performance to that of several other heuristics in practice.

In Section 2 we present the heuristics and state the main results. The proofs appear in Section 3.

We wish to express our gratitude to the referee for numerous helpful comments on the first version of the paper.

2 Heuristics and Theoretical Results

2.1 Random Anticoloring

We start with a naive approach. This approach will lead to a baseline solution, that will be used later to develop a better heuristic. Out of the n vertices of G, we color b arbitrary vertices in black. Coloring now in white every vertex that is not a neighbor of a black vertex, we get an anticoloring of G. This is what we do in Algorithm 1, where the black vertices are selected uniformly randomly (each of the $\binom{n}{b}$ possibilities having the same probability $1/\binom{n}{b}$).

The *neighborhood* $N(v)$ of a vertex $v \in V$ is the set of vertices adjacent to v, and the *closed neighborhood* is $N[v] = N(v) \cup \{v\}$.

2.2 The Performance of Algorithm 1 (Random Anticoloring)

In the following proposition, we will use Pochhammer's symbol (falling factorial)

$$(n)_k = n(n-1) \cdot \dots \cdot (n-k+1), \qquad n, k \geq 0.$$

Algorithm 1 – Random Anticoloring

Input: A graph $G = (V, E)$ and a non-negative integer b.
Output: An anticoloring (B, W) of $G : B$ – black vertices, W – white vertices.

```
procedure RANDOM(G, b)
    B ← ∅
    W ← V
    for i ← 1, 2, ..., b do
        select v_i ∈ V \ B uniformly randomly
        W = W \ N[v_i]
        B = B ⋃ {v_i}
        remove all edges incident to v_i from E
    end for
    return B, W
end procedure
```

We denote by $d(v)$ the degree of a vertex $v \in V$.

The basic measures of the performance of Random Anticoloring are given by

Proposition 1. *Let W be the (random) set of white vertices provided by Algorithm 1. Then:*

(a)
$$E\left(|W|\right) = \frac{1}{(n)_b} \sum_{v \in V} (n - d(v) - 1)_b. \tag{2.1}$$

(b)
$$\begin{aligned} V\left(|W|\right) = &\frac{1}{(n)_b^2} \sum_{v \in V} \left((n - d(v) - 1)_b \cdot (n)_b - (n - d(v) - 1)_b^2\right) \\ &+ \frac{2}{(n)_b} \sum_{(u,v) \in E} (n - d(u) - d(v) + m_{uv})_b \\ &+ \frac{2}{(n)_b} \sum_{(u,v) \notin E} (n - d(u) - d(v) + m_{uv} - 2)_b \\ &- \frac{2}{(n)_b^2} \sum_{(u,v) \in E \cup E^C} (n - d(u) - 1)_b \cdot (n - d(v) - 1)_b, \end{aligned} \tag{2.2}$$

where m_{uv} denotes the number of common neighbors of vertices u and v and E^C is the set of non-edges of G.

While the proof of the proposition will be given in Section 3, it will be important, for the discussion in Subsection 2.3, to understand intuitively the first part of the proposition. In the sum on the right-hand side of (2.1), each summand

$$\frac{(n - d(v) - 1) \cdots (n - d(v) - b)}{n \cdots (n - b + 1)}$$

is actually the probability that neither v nor any of its neighbors will be colored black, so that we will be able to color it white. By denoting

$$P(v) = \frac{(n - d(v) - 1) \cdots (n - d(v) - b)}{n \cdots (n - b + 1)}, \tag{2.3}$$

we may write (2.1) in an alternative way:

$$E\left(|W|\right)=\sum_{v\in V}P(v).$$

Note that $P(v)$ is the same for all vertices of the same degree. Moreover, once we have calculated $P(v)$ for vertices of any degree k, we can find $P(v)$ for vertices of degree $k+1$ in constant time. Thus, we can find $P(v)$ for all $v\in V$ in linear time in n.

2.3 An Improved Heuristic

Proposition 1 provides the average size of W if we choose uniformly randomly b vertices and color them black. We can guarantee this performance by derandomization (1, p.249). Namely, using conditional expectations, we will convert Algorithm 1 into a deterministic heuristic, for which the size of W will be at least the value of the expression on the right-hand side of (2.1). We will improve Algorithm 1 by choosing the black vertices in a different way. Similarly to Algorithm 1, we start with no vertices colored in black, and then color b vertices one by one. Differently from Random Anticoloring, we now color at each step not a random vertex, but rather the "most promising" vertex. Here, by the most promising vertex we mean that vertex which, if colored black, will yield on average the largest number of white vertices at the end of the process. Namely, at each step, for each of the vertices not colored in black, we find the conditional expectation of the size of W at the end of the process if we color this vertex in black and continue the process randomly as in Random Anticoloring.

As in Random Anticoloring, initially we have $B=\emptyset$ and $W=V$. As we continue, the set B is augmented at each step by the vertex we color at that step. The set W is always the set of vertices which may still be white – all vertices neither colored black nor neighboring a black vertex at this stage. Now it is also important to keep track of the set of those vertices which have not been colored in black by this stage, but have a black neighbor, so they will be either black or uncolored by the end of the process. Denote this set of vertices by C. Note that the sets B,W,C are always pairwise disjoint, and together comprise V. Once we keep track of C, we do not need to know exactly which black vertex (or vertices) blocks each vertex in C from being colored white. Thus, at each step, after a vertex has been added to B, we remove from the graph all the edges incident to this vertex, and update the sets B,W,C and the probabilities $P(v)$. Note that $P(v)=0$ for $v\in B\cup C$.

Suppose we have already chosen $i-1$ vertices $v_1,...,v_{i-1}$ for B. The current expected size of W is $E(|W|\mid v_1,v_2,\ldots,v_{i-1})$ (where the condition means that $v_1,\ldots,v_{i-1}\in B$). Let us explain in detail how we select the next vertex to be colored black. To this end, we need to find the effect of adding any candidate $v\in W\cup C$ to B as the next black vertex v_i.

On the one hand, we lose the chance of coloring in white v and all its neighbors (i.e., those vertices of $N[v]$ belonging to W). More specifically, denote by $n'=n-|B|=n-(i-1)$ the current number of non-black vertices in G, and by $b'=b-|B|$ the number of vertices yet to be colored black. Before coloring v, the probability of any vertex u, currently in W, to remain white by the end of the process, if the rest of the coloring is done randomly, is

$$P(u)=\frac{(n'-d(u)-1)\cdots(n'-d(u)-b')}{n'\cdots(n'-b'+1)}. \tag{2.4}$$

Thus, by adding v to B, we reduce the conditional expectation of $|W|$ by

$$\sum_{u \in N[v] \cap W} P(u) = \sum_{u \in N[v] \cap W} \frac{(n' - d(u) - 1) \cdots (n' - d(u) - b')}{n' \cdots (n' - b' + 1)}$$

due to all those vertices of W moving to $B \cup C$.

On the other hand, all vertices of $W \setminus N[v]$ have, if we add v to B, a better chance of remaining white by the end of the run. Namely, $P(u)$ increases for each such vertex u. Indeed, after we have colored v in black, it only remains to color $b' - 1$ vertices out of the remaining non-black $n' - 1$ vertices. Thus, the probability that a vertex $u \in W \setminus N[v]$ will remain white by the end of the run becomes

$$P_{\text{new}}(u) = \frac{n'}{n' - d(u) - 1} \cdot P(u). \tag{2.5}$$

Hence, the expected increase in $|W|$ due to each such vertex u is the difference between the right-hand sides of (2.5) and (2.4)

$$\begin{aligned} &\frac{(n' - d(u) - 2) \cdots (n' - d(u) - b')}{(n' - 1) \cdots (n' - b' + 1)} - \frac{(n' - d(u) - 1) \cdots (n' - d(u) - b')}{n' \cdots (n' - b' + 1)} \\ &= \frac{(n' - d(u) - 2) \cdots (n' - d(u) - b')\,(d(u) + 1)}{n' \cdots (n' - b' + 1)}. \end{aligned}$$

Thus, for $v \in V \setminus B$, we denote

$$g^-(v) = \sum_{u \in N[v] \cap W} P(u), \tag{2.6}$$

and

$$\begin{aligned} g^+(v) &= \sum_{u \in W \setminus N[v]} \frac{(n' - d(u) - 2) \cdots (n' - d(u) - b')\,(d(u) + 1)}{n' \cdots (n' - b' + 1)} \\ &= \sum_{u \in W \setminus N[v]} \frac{d(u) + 1}{n' - d(u) - 1} \cdot P(u). \end{aligned} \tag{2.7}$$

We will refer to $g^-(v)$ and $g^+(v)$ as the *expected negative gain* and the *expected positive gain,* respectively, associated with adding v to B. The *total expected gain* is:

$$g(v) = g^+(v) - g^-(v), \quad v \in V \setminus B. \tag{2.8}$$

At each iteration i, we will act in the same way and color in black the vertex $v_i \in V \setminus B$ such that $g(v_i)$ is maximal among all $g(v)$, $v \in V \setminus B$. (Ties are broken arbitrarily.) After adding v_i to B, we update $P(v)$, and accordingly $g^-(v)$, $g^+(v)$, $g(v)$, for each $v \in V \setminus B$. We call this heuristic *Max Expectation*. See Algorithm 2 for the details.

Algorithm 2 – Max Expectation

Input: A graph $G = (V, E)$, with $|V| = n$, and a non-negative integer b.
Output: An anticoloring of G with b black vertices.

```
procedure MAX EXPECTATION(G, b)
    INITIALIZE(B, W, C, n', b', (P(v), g^-(v), g^+(v), g(v))_{v∈V})
    for i ← 1, 2, ..., b do
        v_i = SELECT VERTEX(W, g(v)_{v∈W})
        COLOR(v_i, B, W, C, n', b', (P(v), g^-(v), g^+(v), g(v))_{v∈V\B})
    end for
    return B, W
end procedure
```

```
procedure INITIALIZE(B, W, C, n', b', (P(v), g^-(v), g^+(v), g(v))_{v∈V})
    B ← ∅
    W ← V
    C ← ∅
    n' ← n
    b' ← b
    for each v ∈ V do
        P(v)   ← the right hand-side of (2.4)
        g^-(v) ← the right hand-side of (2.6)
        g^+(v) ← the right hand-side of (2.7)
        g(v)   ← the right hand-side of (2.8)
    end for
end procedure
```

Theorem 2. *Consider Max Expectation on a graph $G = (V, E)$.*
(i) It finds a legal anticoloring with b black vertices.
(ii) The size $|W|$ of the set of vertices it colors in white is at least the right-hand side of (2.1).
(iii) Its runtime is $O(b \cdot n^2)$.

2.4 The Case of Random Graphs

We will be particularly interested in random graphs. Under the $G(n, p)$ random graph model (9), the graph consists of n labeled vertices, such that the probability of having an edge between any pair of vertices is p, independently of the other pairs. We will deal with sparse graphs: $p = \lambda/n$ for an arbitrary fixed $\lambda > 0$.

We will analyze the performance of Random Anticoloring for $G(n, p)$ graphs, which will yield a lower bound on the performance of Max Expectation. In Proposition 1, the graph was given and the sample space was the set $\mathcal{S}_n$ of all orderings of the graph's vertices, endowed with the uniform distribution. Now the sample space is the product space $\mathcal{G}_n \times \mathcal{S}_n$, where $\mathcal{G}_n$ is the collection of all $2^{n(n-1)/2}$ graphs on n labeled vertices, with the probability measure defined by the $G(n, p)$ model.

Given a sequence $(Y_k)_{k=1}^{\infty}$ of random variables, and a distribution law $\mathcal{L}$, we write $Y_k \xrightarrow[k\to\infty]{\mathcal{D}} \mathcal{L}$ if the sequence converges in distribution to $\mathcal{L}$. We denote by $N(0, 1)$ the standard normal distribution.

---

procedure SELECT VERTEX$((W, g(v)_{v\in W}))$
  $u \leftarrow \arg\max_{v\in W} g(v)$ /* break ties arbitrarily. */
  return u
end procedure

---

procedure COLOR$(u, B, W, C, n', b', (P(v), g^-(v), g^+(v), g(v))_{v\in V\setminus B})$
  $W \leftarrow W \setminus N[u]$
  $B \leftarrow B \bigcup \{u\}$
  $C \leftarrow C \bigcup N(u) \setminus \{u\}$
  remove all edges incident to u from E
  $n' \leftarrow n' - 1$
  $b' \leftarrow b' - 1$
  for $v \in V \setminus B$ **do**
    update $P(v), g^-(v), g^+(v), g(v)$ /*by (2.5), (2.6), (2.7), (2.8), respectively.*/
  end for
end procedure

Proposition 3. *The size of the set W, provided by Random Anticoloring, for a random $G(n, \lambda/n)$ graph and $b = \beta n$, $0 < \beta < 1$, is $B\left(n - b, (1 - \lambda/n)^b\right)$-distributed. In particular:*

(a) $E_{\mathcal{G}_n\times\mathcal{S}_n}(|W|) = (1-\beta)\, e^{-\lambda\beta} n + O(1)$,

(b) $V_{\mathcal{G}_n\times\mathcal{S}_n}(|W|) = (1-\beta)\,(e^{-\lambda\beta} - e^{-2\lambda\beta})n + O(1)$,

(c) $\dfrac{|W| - (1-\beta)\, e^{-\lambda\beta} n}{\sqrt{(1-\beta)\,(e^{-\lambda\beta} - e^{-2\lambda\beta})n}} \xrightarrow[n\to\infty]{\mathcal{D}} N(0,1).$

Remark 1. One can formulate the proposition for general $G(n, p)$ graphs, not necessarily sparse; just replace λ by pn. However, it is easy to see that, for any fixed p and β, the resulting set of white vertices will become very small with high probability as $n \to \infty$. Hence, the interesting random graphs to test our method are sparse.

Recall that, for a sequence $(E_j)_{j=1}^{\infty}$ of events in a probability space, E_j occurs *with high probability* (*w.h.p.*) if $P(E_j) \xrightarrow[j\to\infty]{} 1$. It is well known that (see, for example, (16)) the component structure of $G(n, \lambda/n)$ is:

(i) For $\lambda < 1$, w.h.p. the largest connected component of G is of size $O(\log n)$.

(ii) For $\lambda = 1$, w.h.p. the largest connected component of G is of size $O\left(n^{2/3}\right)$.

(iii) For $\lambda > 1$, w.h.p. there exists a single largest component of G of size $\gamma n\,(1 + o(1))$, where $\gamma \in (0, 1)$ is the unique solution of the equation

$$\gamma + e^{-\lambda\gamma} = 1. \tag{2.9}$$

Moreover, the next largest component in G is of size $O(\log n)$.

Recall that the *Lambert W function* (see, for example, (8, p.241)) is defined implicitly on $[-1/e, \infty)$ by the equation $W(x)\, e^{W(x)} = x$. Note that, in the interval $(-1/e, 0)$, the function has two branches; we take the top (principal) one (see Figure 1).

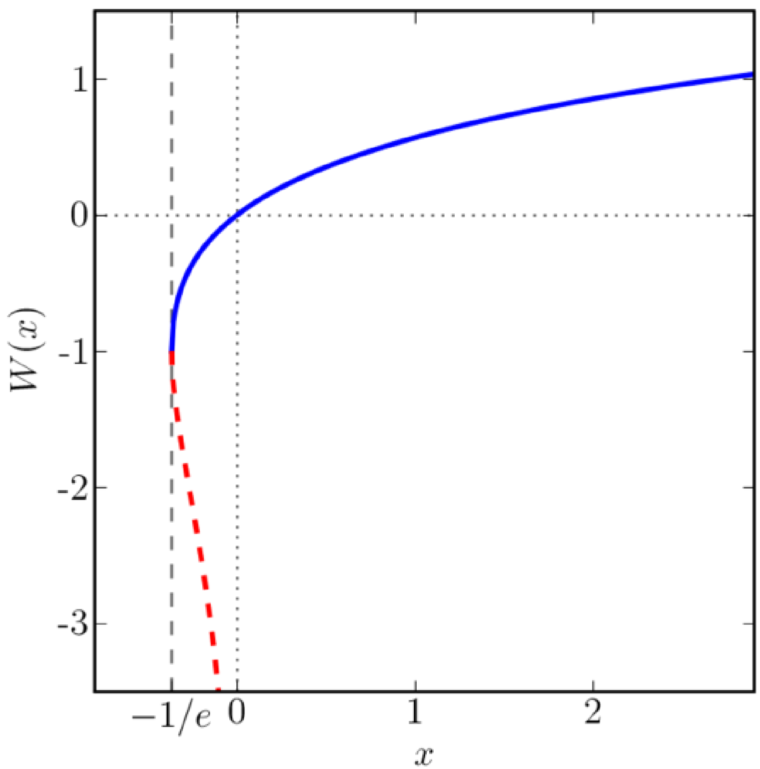

Figure 1: The graph of the Lambert W function. The single-valued function corresponding to the blue solid graph is the *principal branch*, and the dashed red graph is the *negative branch*.

The value of γ solving (2.9) is given by (9):

$$L(\lambda) = \frac{W\left(-\lambda e^{-\lambda}\right) + \lambda}{\lambda}.$$

A trivial upper bound on W_b is $n-b$. When G is not connected, it may be the case that there exists a union of connected components of G with exactly b vertices. In this case we color in black these b vertices and get $n-b$ white vertices. The following theorem guarantees that, for some pairs λ and b, this is possible w.h.p.

Theorem 4. *For $G(n, \lambda/n)$ and $\varepsilon > 0$*
(i) For $\lambda < L^{-1}(1/2)$, w.h.p. we have $W_b = n - b$ for every $0 \le b \le n$.
(ii) For $\lambda \ge L^{-1}(1/2)$ and $\varepsilon > 0$, w.h.p. we have $W_b = n - b$ for every $b \in [0, n(1 - L(\lambda) - \varepsilon)] \bigcup [n(L(\lambda) + \varepsilon), n]$.

Remark 2. In fact, the second part holds for $\lambda < L^{-1}(1/2)$ as well. For such λ we have $L(\lambda) < 1/2$, and therefore

$$[0, n(1 - L(\lambda) - \varepsilon)] \bigcup [n(L(\lambda) + \varepsilon), n] = [0, n].$$

We preferred to state this case more explicitly in part (i).

The proof of Theorem 4 is algorithmic. One starts with $B = \emptyset$, and adds to it at each step the largest possible connected component. See Algorithm 3 for details.

Algorithm 3 – Largest Connected Components

Input: A graph $G = (V, E)$ and a non-negative integer b.
Output: An anticoloring of G with b black vertices.

```
procedure LARGEST CONNECTED COMPONENTS(G, b)
    B ← ∅
    W ← V
    Sort the connected components of G according to descending order of size: |V_1| ≥ |V_2| ≥ ··· ≥ |V_k|.
    for i ← 1, 2, ..., k do
        if |B| + |V_i| ≤ b then
            B = B ⋃ V_i
            W = W \ V_i
        end if
    end for
    return B, W
end procedure
```

Knowing the performance of any anticoloring heuristic, we obtain a lower bound on the size of the optimal W. Thus, Proposition 3 guarantees that the optimum is at least $(1-\beta)e^{-\lambda\beta}n + O(1)$ for $G(n, \lambda/n)$ with $b = \beta n$ black vertices w.h.p. The bound is based on the performance of the trivial solution, obtained by coloring b random vertices in black. Algorithm 2 would provide a better bound, but we do not know how to find its typical performance. A simple heuristic, starting from the same basic idea, is the following greedy heuristic. Note that, when coloring vertices in black one by one, as we color some vertex v, we block all vertices of $N[v]$, not adjacent to any vertex already colored, from being eventually colored in white. Thus, it is natural to color in each step a vertex v for which $|N[v] \cap W|$ is minimal. See Algorithm 4 for details.

Unfortunately, even Algorithm 4 is not readily amenable to analysis. Hence, we take a similar, but even simpler heuristic. We simply color in black the b vertices of minimal degrees. More precisely, denote by A_k the set of all vertices of degree k, $0 \le k \le n-1$. Let t be the largest number such that the number of vertices of degree up to t does not exceed b:

$$\sum_{k=0}^{t} |A_k| \le b < \sum_{k=0}^{t+1} |A_k|. \tag{2.10}$$

We color in black all the vertices of $\bigcup_{k=0}^{t} A_k$, and $b - \sum_{k=0}^{t-1} |A_k|$ additional random vertices out of A_{t+1}. See Algorithm 5 for the details.

Theorem 5 provides the average performance of Algorithm 5, and thus gives a lower bound on the average value of W_b.

Algorithm 4 – Greedy

Input: A graph $G = (V, E)$ and a non-negative integer b.
Output: An anticoloring of G with b black vertices.

```
procedure GREEDY(G, b)
    B ← ∅
    W ← V
    for i ← 1, 2, ..., b do
        v_i ← argmin_{v∈V} |N[v] ∩ W|        /* break ties arbitrarily. */
        B = B ⋃ {v_i}
        W = W \ N[v_i]
        remove all edges incident to v_i from E
        remove v_i from V
    end for
    return B, W
end procedure
```

Algorithm 5 – Min Degree

Input: A graph $G = (V, E)$ and a non-negative integer b.
Output: An anticoloring of G with b black vertices.

```
procedure MIN DEGREE(G, b)
    Sort the vertices of G according to ascending order of degrees:
    d(v_1) ≤ d(v_2) ≤ ··· ≤ d(v_n).
    /* between vertices of the same degree, the order is uniformly random. */
    B = {v_1, v_2, ..., v_b}
    W = V \ ⋃_{i=1}^{b} N[v_i]
    return B, W
end procedure
```

Theorem 5. *For $G(n,\lambda/n)$ and $b=\beta n$, where $0<\beta<1$ is fixed and $n\to\infty$, let W be the (random) set of white vertices provided by Min Degree. Define the integer t by the inequalities*

$$\sum_{k=0}^{t-1}\frac{e^{-\lambda}\lambda^k}{k!}\le\beta<\sum_{k=0}^{t}\frac{e^{-\lambda}\lambda^k}{k!},$$

and let $r=\sum_{k=0}^{t}\frac{e^{-\lambda}\lambda^k}{k!}-\beta$, $s=\frac{e^{-\lambda}\lambda^t}{t!}$. Then:

$$E_{\mathcal{G}_n}\left(|W|\right)=\begin{cases} n\left(1-\beta\right)+O(1), & t=0,\\ n\cdot\left[\sum\limits_{k=t+1}^{\infty}\frac{e^{-\lambda}\lambda^k}{k!}\left(\frac{r}{s}\cdot\frac{e^{-\lambda}\lambda^{t-1}}{(t-1)!}+\sum\limits_{j=t}^{\infty}\frac{e^{-\lambda}\lambda^j}{j!}\right)^k\right. & \\ \qquad\left.+r\cdot\left(\frac{r}{s}\cdot\frac{e^{-\lambda}\lambda^{t-1}}{(t-1)!}+\sum\limits_{j=t}^{\infty}\frac{e^{-\lambda}\lambda^j}{j!}\right)^t\right]+O(1), & t\ge 1.\end{cases} \tag{2.11}$$

In the next theorem we present an upper bound on W_b.

Theorem 6. *For $G(n,\lambda/n)$ and $b=\beta n$, where*

$$\lambda>\frac{-\beta\log\beta-(1-\beta)\log\left(1-\beta\right)}{\beta\left(1-\beta\right)}, \tag{2.12}$$

w.h.p. we have $W_b\le\alpha_0 n$, where α_0 is the unique solution of the equation

$$\alpha\log\alpha+\beta\log\beta+(1-\alpha-\beta)\log\left(1-\alpha-\beta\right)+\lambda\alpha\beta=0 \tag{2.13}$$

in the interval $(0,1-\beta)$.

We have run a simulation to test the performance of Greedy and Max Expectation. We tested $G(1000,\lambda/1000)$ graphs for $\lambda=0.1,0.2,\ldots,10.0$. The proportion of black vertices was $\beta=1/4$ for each graph. We took 100 random instances for each λ, and ran both heuristics on these instances. In Figure 2 we depict the proportion of white vertices for each heuristic over these instances. In addition, we present the theoretical prediction for the performance of Random Anticoloring, given in Proposition 3.(a), as well as the lower bound of Theorem 5 and the upper bound of Theorem 6. Max Expectation achieved better results than Min Greedy for all values of λ, which in turn performed better than Degree. The theoretical prediction for the performance of Random Anticoloring was worse than all, and the upper bound above them all.

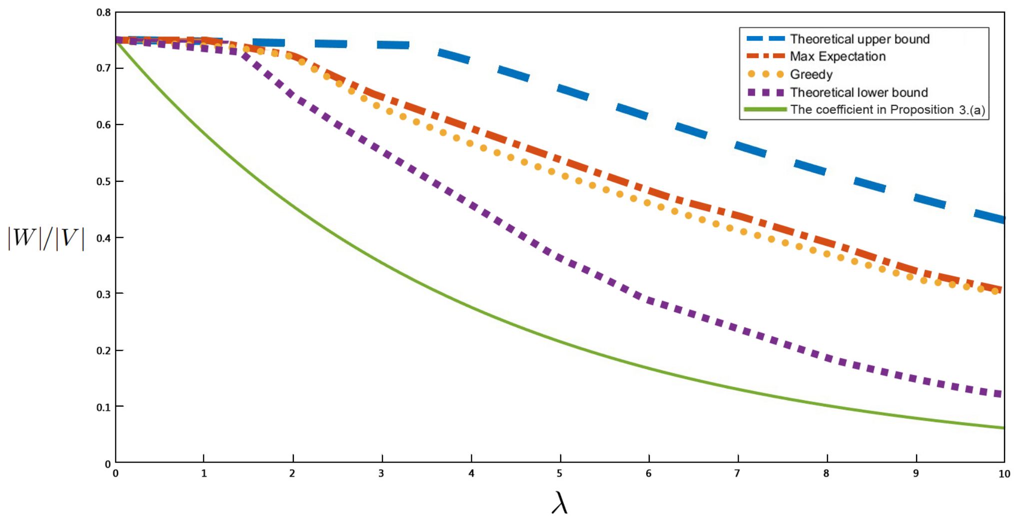

Figure 2: The performance of Greedy and Max Expectation vs. Random Anticoloring and theoretical bounds.

3 Proofs

Proof of Proposition 1. Let $V = \{v_1, v_2, \dots, v_n\}$ and let

$$X_i = \begin{cases} 1, & v_i \in W, \\ 0, & \text{otherwise}, \end{cases} \qquad 1 \le i \le n. \tag{3.1}$$

It is easy to see that $X_i \sim \text{Ber}\left(\frac{(n-d(v_i)-1)_b}{(n)_b}\right)$, and

$$|W| = X_1 + X_2 + \cdots + X_n. \tag{3.2}$$

(a) We have:

$$E\left(|W|\right) = \sum_{i=1}^{n} E\left(X_i\right) = \frac{1}{(n)_b} \sum_{i=1}^{n} (n-d(v_i)-1)_b.$$

(b) By (3.2),

$$\begin{aligned} V\left(|W|\right) &= V\left(\sum_{i=1}^{n} X_i\right) \\ &= \sum_{i=1}^{n} V\left(X_i\right) + 2\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} \text{Cov}\left(X_i, X_j\right). \end{aligned} \tag{3.3}$$

Clearly,

$$\begin{aligned} V\left(X_i\right) &= \frac{(n-d(v_i)-1)_b}{(n)_b} \cdot \left(1 - \frac{(n-d(v_i)-1)_b}{(n)_b}\right) \\ &= \frac{(n-d(v_i)-1)_b \cdot (n)_b - (n-d(v_i)-1)_b^2}{(n)_b^2}, \qquad 1 \le i \le n. \end{aligned} \tag{3.4}$$

It remains to calculate $\mathrm{Cov}(X_i, X_j) = E(X_iX_j) - E(X_i)E(X_j)$. We have

$$E(X_iX_j) = P(X_i = X_j = 1).$$

If $(v_i, v_j) \in E$, then

$$P(X_i = X_j = 1) = \binom{n - (d(v_i) + d(v_j)) + m_{ij}}{b} \Big/ \binom{n}{b},$$

and therefore:

$$\begin{aligned} \mathrm{Cov}(X_i, X_j) = & \frac{(n - d(v_i) - d(v_j) + m_{ij})_b}{(n)_b} \\ & - \frac{(n - d(v_i) - 1)_b}{(n)_b} \cdot \frac{(n - d(v_j) - 1)_b}{(n)_b}. \end{aligned} \tag{3.5}$$

Now let $(v_i, v_j) \notin E$. Then

$$P(X_i = X_j = 1) = \binom{n - (d(v_i) + d(v_j) + 2) + m_{ij}}{b} \Big/ \binom{n}{b}.$$

Therefore, for $(v_i, v_j) \notin E$:

$$\begin{aligned} \mathrm{Cov}(X_i, X_j) = & \frac{(n - d(v_i) - d(v_j) + m_{ij} - 2)_b}{(n)_b} \\ & - \frac{(n - d(v_i) - 1)_b}{(n)_b} \cdot \frac{(n - d(v_j) - 1)_b}{(n)_b}. \end{aligned} \tag{3.6}$$

Our claim follows from (3.3), (3.4), (3.5) and (3.6). □

Proof of Theorem 2. (i) According to the heuristic, at each step, after a vertex v_i has been added to B, we remove from the graph all the edges incident to this vertex, and all its neighbors which belong to W move to C. During the run, no vertices are added to W, so there is no black vertex that has a white neighbor.
(ii) In this process, at each iteration the expected size of W does not decrease. In fact, if we add to B a vertex $v_i = \arg\max_{v \in V \setminus B} g(v)$, the expected size of W will increase by $g(v_i)$. Clearly, $\sum_{v \in V \setminus B} g(v) = 0$, so it is impossible that $g(v) < 0$ for all $v \in V \setminus B$. (We note that it is possible, though, that at some point during the run we have $g(v) = 0$ for all $v \in V \setminus B$. In this case, we add to B any $v \in V \setminus B$, and the expected size of W does not change.)
(iii) Most of the time is spent on the b calls to Color. Within this procedure, most of the time is spent on the updates of $g^-(v)$ and $g^+(v)$. Each update takes $O(n)$ time for each vertex, and we have to deal with $|V - B| \le n$ vertices. Hence the runtime is $O(b \cdot n^2)$. □

In the proofs of the theorems relating to sparse random graphs, we use the Poissonian approximation to the binomial distribution. Let the random variable D_i denote the degree of v_i. Clearly, $D_i \sim B(n-1, \lambda/n)$. By the Poissonian approximation, D_i is distributed approximately $\mathrm{Po}(\lambda)$. More precisely,

$$P(D_i = k) = \frac{e^{-\lambda}\lambda^k}{k!} + \varepsilon_k,$$

where, by (12, p.187),

$$\sum_{k=0}^{\infty} | \varepsilon_k | \leq \frac{4\lambda}{n}.$$

As is usually done, in most proofs we will treat D_i as exactly Poissonian. The involved error is easily seen to be absorbed in the big Oh term.

Proof of Proposition 3. By symmetry, we may assume that we take n vertices $\{v_1, v_2, \dots, v_n\}$, color in black the first b of them, so that $B = \{v_1, v_2, \dots, v_b\}$, and then select the edges/ non-edges randomly. The probability of each of the remaining vertices $v_{b+1}, \dots, v_n$ to belong to W is $(1-\lambda/n)^b$, and the respective events are independent. Hence, $|W| \sim B\left(n-b, (1-\lambda/n)^b\right)$. Thus:

(a) $E(|W|) = (n-b)(1-\lambda/n)^b = n(1-\beta)e^{-\lambda\beta} + O(1)$.

(b)
$$\begin{aligned} V(|W|) &= (n-b)(1-\lambda/n)^b(1-(1-\lambda/n)^b) \\ &= n(1-\beta)(e^{-\lambda\beta}e^{-2\lambda\beta}) + O(1). \end{aligned}$$

(c) Follows from (a), (b), and the Central Limit Theorem.

□

Proof of Theorem 4. (i) We use Algorithm 3. If we have colored in black all the components of size larger than 1, and the process is not over, we continue to color in black components of size 1, until we color b vertices. If we have skipped along the way at least one of the components of size larger than 1, it means that there remain to color in black less than $O(\log n)$ vertices. By [1, Theorem 3.1, p.52], the number of isolated vertices in $G(n, \lambda/n)$ is approximately $ne^{-\lambda}$. Since $ne^{-\lambda} = \omega(\log n)$, we can find a union of components of G with exactly b vertices, and therefore $W_b = n - b$.

(ii) If $b \in [0, n(1-L(\lambda)-\varepsilon)]$ then $b \leq |\bigcup_{i=2}^{k} V_i|$. By using Algorithm 3, we skip over V_1. If we have colored in black all the components of size larger than 1, and the process is not over, we continue to color in black components of size 1, until we color b vertices. If we have skipped another component of size larger than 1, then according to the argument we mentioned in the first part, we can find a union of components of G with exactly b vertices.

If $b \in [n(L(\lambda)+\varepsilon), n]$ then $b \geq |V_1|$. We color V_1 in black. If we did not skip on one of the components of size larger than one, and the process is not over yet, then we will get to components of size one, and we can find an union of components of G with exactly b vertices. If we skipped on one of the components of size larger than one, then according to the argument we mentioned in the first part, we can find an union of components of G with exactly b vertices. □

Proof of Theorem 5. Let $X_i, 1 \leq i \leq n$, be as in (3.1). For

$$\sum_{k=0}^{t-1} \frac{e^{-\lambda}\lambda^k}{k!} < \beta < \sum_{k=0}^{t} \frac{e^{-\lambda}\lambda^k}{k!},$$

w.h.p. all the vertices $v \in V$ with $d(v) < t$ are colored black by the end of the run, while vertices with $d(v) > t$ are not. Some of the vertices $v \in V$ with $d(v) = t$ are colored black, and some are not. Hence, a vertex $v \in V$ with

- $d(v) < t$ – does not belong to W,
- $d(v) = t$ – belongs to W if it is neither black nor a neighbor of a black vertex,
- $d(v) > t$ – belongs to W if it is not a neighbor of a black vertex.

The probability that a vertex $v \in V$ is of degree exceeding t is $\sum_{k=t+1}^{\infty} \frac{e^{-\lambda}\lambda^k}{k!}$. Suppose that $d(v) = k$. The probability that a neighbor u of v is of some degree $j \geq 1$ is about $e^{-\lambda}\lambda^{j-1}/(j-1)!$. The probability of such a vertex u being non-black is therefore

$$\frac{r}{s} \cdot \frac{e^{-\lambda}\lambda^{t-1}}{(t-1)!} + \sum_{j=t}^{\infty} \frac{e^{-\lambda}\lambda^j}{j!}.$$

Hence the probability of a vertex v being of degree exceeding t and white is

$$\sum_{k=t+1}^{\infty} \frac{e^{-\lambda}\lambda^k}{k!} \left(\frac{r}{s} \cdot \frac{e^{-\lambda}\lambda^{t-1}}{(t-1)!} + \sum_{j=t}^{\infty} \frac{e^{-\lambda}\lambda^j}{j!} \right)^k. \tag{3.7}$$

Similarly, the probability that a vertex v of degree t is white is

$$r \cdot \left(\frac{r}{s} \cdot \frac{e^{-\lambda}\lambda^{t-1}}{(t-1)!} + \sum_{j=t}^{\infty} \frac{e^{-\lambda}\lambda^j}{j!} \right)^t. \tag{3.8}$$

The result follows from (3.7) and (3.8). □

Proof of Theorem 6. Let B and W be disjoint sets of sizes βn and αn, respectively, where $0 < \alpha < 1 - \beta$. We upbound the probability that B and W may serve as the required sets of black and of white vertices, namely that there are no edges connecting the two sets. Draw the edges of the graph according to the $G(n, \lambda/n)$ model. The probability that $(u, v) \notin E$ for every $u \in B, v \in W$ is $(1-\lambda/n)^{n^2\alpha\beta}$. The number of all possible choices of these sets B and W is $\binom{n}{\alpha n, \beta b, n(1-\alpha-\beta)}$. By the union bound,

$$P(W_b \geq \alpha n) \leq \binom{n}{\alpha n, \beta n, n(1-\alpha-\beta)} (1-\lambda/n)^{n^2\alpha\beta}.$$

By Stirling's Formula (15),

$$\sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n} e^{\frac{1}{12n+1}} < n! < \sqrt{2\pi} n^{n+\frac{1}{2}} e^{-n} e^{\frac{1}{12n}}.$$

A routine calculation yields

$$\binom{n}{\alpha n, \beta b, n(1-\alpha-\beta)} (1-\lambda/n)^{n^2\alpha\beta} < \frac{c}{n} \cdot \left(\alpha^{-\alpha}\beta^{-\beta} (1-\alpha-\beta)^{\alpha+\beta-1} \right)^n \cdot (1-\lambda/n)^{n^2\alpha\beta},$$

where $c = 1/\left(2\pi\sqrt{\alpha\beta\left(1-\alpha-\beta\right)}\right)$.

Define $g : (0, 1-\beta) \longrightarrow \mathbf{R}$ by:

$$g\left(\alpha\right) = \alpha\log\alpha + \beta\log\beta + \left(1-\alpha-\beta\right)\log\left(1-\alpha-\beta\right), \qquad 0 < \alpha < 1-\beta.$$

Since $\log\left(1-\lambda/n\right) < -\lambda/n$,

$$\begin{aligned}\left(\alpha^{-\alpha}\beta^{-\beta}\left(1-\alpha-\beta\right)^{\alpha+\beta-1}\right)^{n}\cdot\left(1-\lambda/n\right)^{n^2\alpha\beta} &< e^{-n\left(g(\alpha)+n\alpha\beta\cdot(\lambda/n)\right)} \\ &= e^{-n\left(g(\alpha)+\lambda\alpha\beta\right)}.\end{aligned} \tag{3.9}$$

Define $f : [0, 1-\beta] \longrightarrow \mathbf{R}$ by:

$$f\left(\alpha\right) = \begin{cases}\beta\log\beta + \left(1-\beta\right)\log\left(1-\beta\right), & \alpha = 0, \\ \alpha\log\alpha + \beta\log\beta + \left(1-\alpha-\beta\right)\log\left(1-\alpha-\beta\right) + \lambda\alpha\beta, & 0 < \alpha < 1-\beta, \\ \left(1-\beta\right)\log\left(1-\beta\right) + \beta\log\beta + \lambda\left(1-\beta\right)\beta, & \alpha = 1-\beta.\end{cases}$$

(Here and below, refer to Figure 3.) Then,

$$f'\left(\alpha\right) = \log\alpha - \log\left(1-\alpha-\beta\right) + \lambda\beta, \qquad 0 < \alpha < 1-\beta,$$

and $\lim_{\alpha\to 0^+} f'\left(\alpha\right) = -\infty$, $\lim_{\alpha\to(1-\beta)^-} f'\left(\alpha\right) = \infty$. Note that

$$f''\left(\alpha\right) = \frac{1-\beta}{\alpha\left(1-\alpha-\beta\right)}, \qquad 0 < \alpha < 1-\beta,$$

so that $f''\left(\alpha\right) > 0$ for all $0 < \alpha < 1-\beta$. Therefore, f' is an increasing function. Denote by α_1 the unique solution of $f'\left(\alpha\right) = 0$. Thus, f decreases in $[0, \alpha_1]$ and increases in $[\alpha_1, 1-\beta]$. Notice that $f(0) < 0$ and $f(1-\beta) > 0$ for

$$\lambda > \frac{-\left(1-\beta\right)\log\left(1-\beta\right) - \beta\log\beta}{\beta\left(1-\beta\right)}.$$

Hence, there is indeed a unique α_0 such that $f(\alpha_0) = 0$, namely an α_0 solving (2.13). For $\alpha > \alpha_0$ we have $f(\alpha) > 0$, so that the value of the right-hand side of (3.9) goes to 0 as $n \to \infty$. Thus, the probability that $W_b > \alpha_0 n$ goes to 0 as $n \to \infty$. □

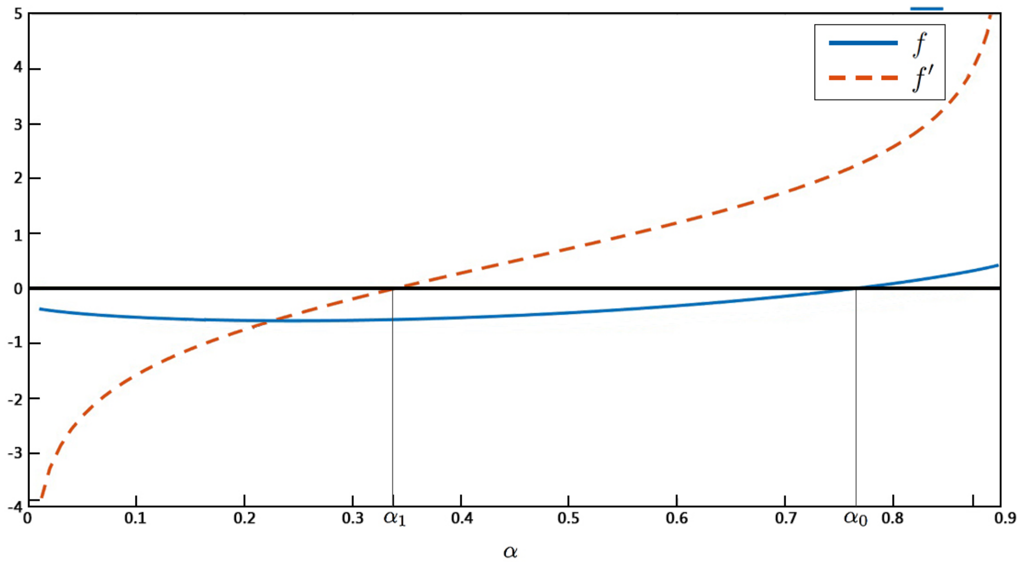

Figure 3: The functions f and f' for $\beta = 0.1, \lambda = 10$.

References

[1] N. Alon and J. Spencer. *The Probabilistic Method.* Wiley, New York, 1992.

[2] D. Berend, E. Korach, and O. Yahalom. Anticoloring of the rook's graph. *Discrete Applied Mathematics*, 188:1–15, 2015. doi:10.1016/j.dam.2015.02.023.

[3] D. Berend, E. Korach, and S. Zucker. A reduction of the anticoloring problem to connected graphs. *Electronic Notes in Discrete Mathematics*, 28:445–451, 2007. 6th Czech-Slovak International Symposium on Combinatorics, Graph Theory, Algorithms and Applications. doi:10.1016/j.endm.2007.01.062.

[4] D. Berend, E. Korach, and S. Zucker. Anticoloring of a family of grid graphs. *Discrete Optimization*, 5:647–662, 2008. doi:10.1016/j.disopt.2008.02.001.

[5] D. Berend, E. Korach, and S. Zucker. Anticoloring and separation of graphs. *Discrete Mathematics*, 310:390–399, 2010. doi:10.1016/j.disc.2009.03.027.

[6] D. Berend, E. Korach, and S. Zucker. Tabu search for the bwc problem. *Journal of Global Optimization*, 54:649–667, 2012. doi:10.1007/s10898-011-9783-1.

[7] D. Berend and S. Zucker. The black-and-white coloring problem on trees. *Journal of Graph Algorithms and Applications*, 13:133–152, 2009. doi:10.7155/jgaa.00180.

[8] B. Bollobás. *Modern Graph Theory.* Springer-Verlag, New York, NY, 1998. doi:10.1007/978-1-4612-0619-4.

[9] B. Bollobás. *Random Graphs.* Cambridge University Press, Cambridge, 2nd edition, 2001. doi:10.1017/CBO9780511814068.

[10] P. Hansen, A. Hertz, and N. Quinodoz. Splitting trees. *Discrete Mathematics*, 165/166:403–419, 1997.

[11] D. Kobler, E. Korach, and A. Hertz. On black-and-white colorings, anticolorings and extensions. Preprint.

[12] J. Neyman and L. L. Cam. On the distribution of sums of independent random variables. In *Bernoulli, Bayes, Laplace. Anniversary Volume*, pages 179–202. Springer-Verlag, 1965.

[13] L. E. U. Rivero, J. R. Rodriguez, and R. L. Bracho. Balanced black and white coloring problem on knight's chessboards. In *15th International Conference on Electrical Engineering, Computing Science and Automatic Control (CCE)*, pages 1–5, 2018.

[14] L. E. U. Rivero, J. R. Rodríguez, and R. L. Bracho. Genetic algorithm for black and white coloring problem on graphs. *Programación Matemática y Software*, 10(1):17–23, 2018.

[15] H. Robbins. A remark on stirling's formula. *The American Mathematical Monthly*, 62:26–29, 1955.

[16] C. Witt. Analysis of an iterated local search algorithm for vertex cover in sparse random graphs. *Theoretical Computer Science*, 425:117–125, 2012. doi:10.1016/j.tcs.2011.01.010.

[17] S. Zucker. The black-and-white coloring problem on chordal graphs. *Journal of Graph Algorithms and Applications*, 16(2):261–281, 2012. doi:10.7155/jgaa.00258.

[18] S. Zucker. The d-distance anticoloring problem. In *The conference Computability in Europe (CiE), The Nature of Computation*, volume 251, 2013.