

# Homework Report

Shawn Seymour  
University of Minnesota Morris

Last modified: January 31, 2017

## Heuristics

In this report, I will discuss multiple vertex coloring heuristics. The input for all heuristics defined here is a simple graph  $G = (V, E)$ . I will define them as follows:

### Heuristic A

Heuristic A is the greedy algorithm.

---

**Algorithm 1** Greedy algorithm

---

- 1: Label each vertex in  $V$ , i.e.  $v_1, v_2, \dots, v_n$
  - 2: **for** each  $v \in V$  **do**
  - 3:     Assign a color  $p_i$  to  $v_i$  using the smallest available  $p_i$
- 

### Heuristic B

Heuristic B is the greedy algorithm with degree sequencing. It orders the vertices according to the decreasing value of their degree. This is also known as the Welsh-Powell algorithm, which is defined in [7].

---

**Algorithm 2** Welsh-Powell algorithm

---

- 1: Label each vertex in  $V$ , i.e.  $v_1, v_2, \dots, v_n$ , such that  $d_G(v_1) \geq d_G(v_2) \geq \dots \geq d_G(v_n)$
  - 2: **for all**  $v \in V$  **do**
  - 3:     Assign a color  $p_i$  to  $v_i$  using the smallest available  $p_i$
- 

### Heuristic C

Heuristic C colors a graph by finding maximal independent sets of  $G$ .

---

**Algorithm 3** Coloring via maximal independent set algorithm

---

- 1: **while**  $G$  is non-empty **do**
  - 2:     Find a maximal independent set of  $G$ , i.e.  $S_i$
  - 3:     Color all vertices in  $S_i$  with color  $p$ , where  $p$  is the smallest color available
  - 4:     Let  $G \leftarrow G[V \setminus S_i]$
-

## Findings

**Proposition 1.** *Heuristic A and Heuristic B do not always produce an optimal solution, i.e. they do not always produce a minimum coloring of a graph.*

I will show this for *heuristic B*. It is trivial to show the same for *heuristic A* as *heuristic B* includes the same steps as *heuristic A*. I will construct a simple graph,  $G = (V, E)$ , such that:

$$|V| \geq 8 \tag{1}$$

$$\chi(G) > \chi^*(G) \tag{2}$$

Let  $\chi$  refer to the coloring of  $G$  generated by *heuristic B*. Let  $\chi^*$  be the optimal coloring of  $G$ .

### Example 1

I constructed a simple graph,  $G = (V, E)$ , such that  $\Delta(G) = 3$  and  $\delta(G) = 1$ . I've let  $|V| = 8$  for this example.

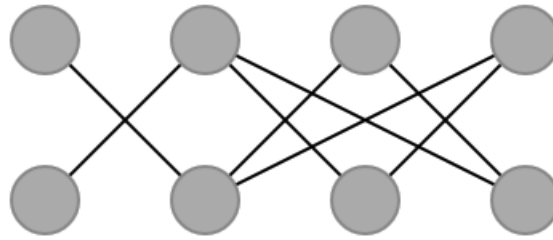


Figure 1: Uncolored original graph  $G$

By applying *heuristic B*, we get the following coloring. The numbers indicate the ordering of vertices before applying the heuristic. Any vertex of the same degree got assigned arbitrarily. This results in  $\chi(G) = 3$ .

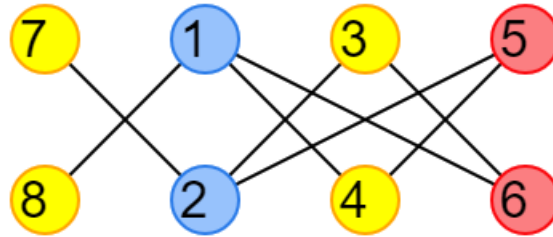


Figure 2: Coloring from applying heuristic B

**Definition** (Bipartite graph). *A bipartite graph is one whose vertex set can be partitioned into two subsets  $X$  and  $Y$  such that each edge has one end in  $X$  and one end in  $Y$*

We can see that this is a bipartite graph, defined above by [1]. Thus,  $G$  is *2-colorable*. This means  $\chi^*(G) = 2$ .  $G$  is an example graph that satisfies conditions (1) and (2). The optimal coloring is shown below.

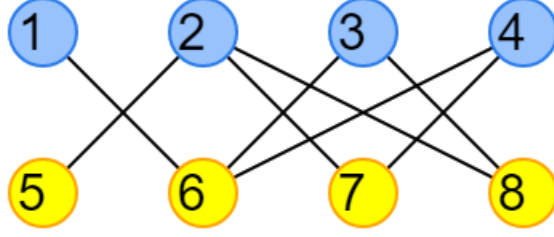


Figure 3: Optimal coloring of graph  $G$

**Proposition 2.** *Heuristic C also does not always produce an optimal solution, i.e. it does not always produce a minimum coloring of a graph.*

### Example 2

To show this, I will create another example that satisfies conditions (1) and (2) from above. I've constructed a graph  $K = (V, E)$  for this example.

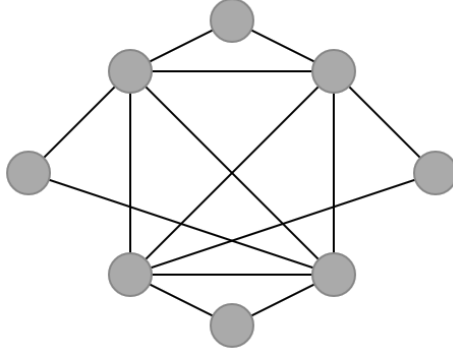


Figure 4: Uncolored original graph  $K$

By applying heuristic C and finding maximal independent sets, we see that  $K$  is *5-colorable*.

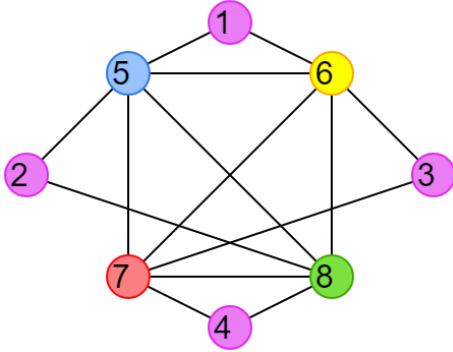


Figure 5: Coloring of  $K$  after applying heuristic C

This is a non-optimal solution. We can see that the subgraph created by vertices  $\{5, 6, 7, 8\}$  is complete and thus we need at least 4 colors. This graph is *4-colorable* however, meaning  $\chi^*(K) = 4$ . Thus, we can see that heuristic C does not always give us an optimal solution.

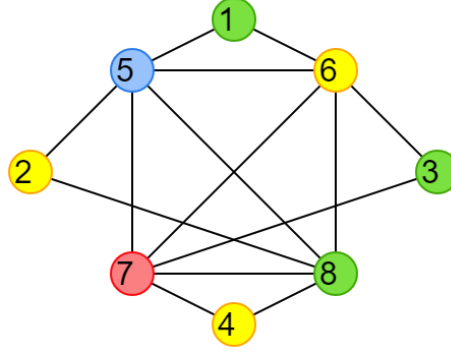


Figure 6: Optimal coloring of  $K$

**Proposition 3.** *Heuristic A and heuristic B have an upper bound of  $\Delta + 1$ .*

To show this, I will create another example that satisfies conditions (1) and (2) from above. After doing some research into bipartite graphs, I learned that *crown graphs* are excellent at showing how bad greedy heuristics can be, as shown and defined in [4].

**Definition** (Crown Graph). *A crown graph  $CR_n = (V, E)$  is an undirected graph with two sets of vertices where  $V = V_1 \cup V_2$  with an edge from  $v_i \in V_1$  to  $v_j \in V_2$  whenever  $i \neq j$ . A crown graph can also be viewed as a complete bipartite graph where the edges of a perfect matching have been removed.*

### Example 3

I constructed a simple crown graph,  $H = (V, E)$ . I've let  $|V| = 10$  for this example.

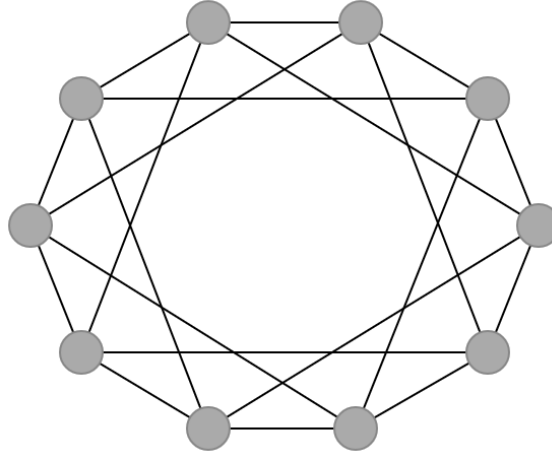


Figure 7: Uncolored original graph  $H$

We can see that  $\Delta(G) = 4$ . We can also see  $d_G(v) = 4$  for all  $v \in V$ . Thus, in both heuristics A and B, the greedy algorithm would pick an order arbitrarily. We can show using this crown graph the worst-case scenario of these heuristics.

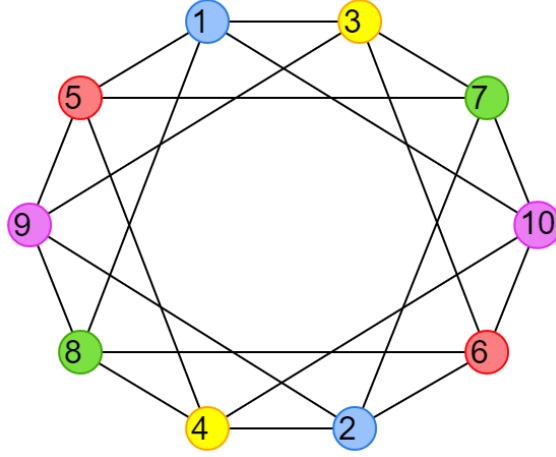


Figure 8: Worst-case coloring of  $H$  using either heuristic A or B

In Figure 5, using either heuristic with this ordering, we get  $\chi(H) = 5$ . This gives us  $\frac{|V|}{2}$  colors. This is the worst-case for a crown graph as shown in [2], but this graph also demonstrates the upper bounds for these heuristics.

**In General:** Let's take a look at how this looks in general. Let  $P = (V, E)$  be a simple, complete graph.

We can see that  $\chi(H) = \Delta(H) + 1$ . This is very easy to see in a *complete* graph, where all vertices are connected to every other vertex. This means all vertices have degree  $|V| - 1$ . Thus, every time we color a node, a new color is needed. And since we have  $\Delta(P) = |V| - 1$  and  $|V|$  vertices, we will need  $\Delta(P) + 1$  colors. This is stated in *Brooks' Theorem*.

**Theorem** (Brooks' Theorem). *For any connected undirected graph  $G$  with maximum degree  $\Delta$ , the chromatic number of  $G$  is at most  $\Delta$  unless  $G$  is a complete graph or an odd cycle, in which case the chromatic number is  $\Delta + 1$ .*

The proof of *Brook's Theorem* can be found in [6]. Overall, heuristic A and heuristic B can produce some very undesirable results. In graph  $H$ , at the worst case, these heuristics produce  $\chi(H) = 5$  when  $\chi^*(H) = 2$  as it is bipartite. This is shown below.

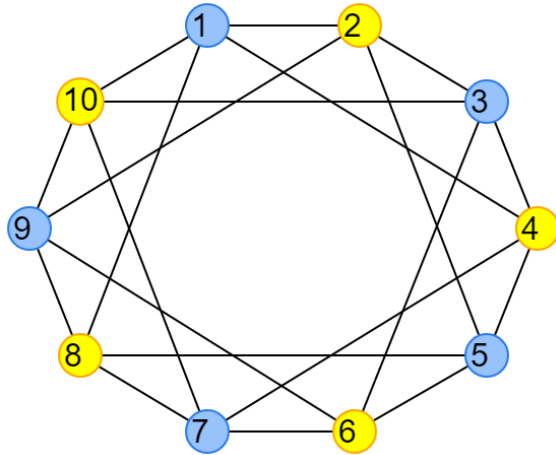


Figure 9: Optimal coloring of  $H$

**Proposition 4.** *Heuristic A and Heuristic B have a running time of  $O(|V| + |E|)$ .*

It is possible to implement heuristic A and heuristic B efficiently and achieve a linear running time [5]. By using an adjacency list representation of a graph  $G = (V, E)$ , the greedy algorithm will look once at each vertex and each edge twice. This is because for each vertex, all of its neighbors are checked. This leads to the algorithm crossing each edge twice. As this is still linear, we know  $O(|V| + 2|E|) = O(|V| + |E|)$ .

**Proposition 5.** *Heuristic A and Heuristic B produce a feasible, proper coloring of input graph  $G = (V, E)$ .*

Let  $G = (V, E)$  be a simple graph. Assume that two vertices, say  $v_1, v_2 \in V$ , are connected by an edge  $e_1 \in E$ . Assume that  $v_1, v_2$  are colored the same color, say  $p_1$ . We know that in both heuristics, we assign the smallest color  $p_i$  that is not being used by any of the neighboring vertices connected by an edge  $e_i$ . WLOG, the heuristic would color  $v_1$  color  $p_1$ . The next iteration, while vertex  $v_2$  is selected, the heuristic would see color  $p_1$  is assigned to a neighboring vertex ( $v_1$ ), and assign the next available color  $p_2$ . Thus, we've reached a contradiction.

## Complexity of Vertex Coloring

**Proposition 6.** *Vertex coloring is an NP-hard problem.*

Coloring a graph is computationally complex. It is NP-hard to compute the chromatic number of a given graph [3]. It is NP-complete to decide if a given graph has a  $k$ -coloring for a given  $k$ . This is because it is easy to verify if a coloring of the given graph is a  $k$ -coloring in polynomial time. This can be done by iterating through the list of edges and checking that each attached vertex has a different color.

## References

- [1] Bondy, J.A. and U.S.R. Murty [1976], *Graph theory with applications*, American Elsevier Publishing, New York, NY.
- [2] Johnson, D. S. [1974], *Worst-case behavior of graph coloring algorithms*, Proc. 5th Southeastern Conf. on Combinatorics, Graph Theory, and Computing, Utilitas Mathematicae, Winnipeg, pp. 513–527.
- [3] Garey, M. R., Johnson, D. S., & Stockmeyer, L. [1976], *Some simplified NP-complete graph problems*, Theoretical computer science, 1(3), 237-267.
- [4] Kordecki, W. and A. Łyczkowska-Hanćkowiak [2016], *Greedy online colouring with buffering*, arXiv preprint arXiv:1601.00252.
- [5] Kubale, Marek [2004], *Graph colorings*, Vol. 352, American Mathematical Soc.
- [6] Lovász, L. [1975], *Three short proofs in graph theory*, Journal of Combinatorial Theory, Series B, 19(3), 269-271.
- [7] Welsh, D. J. and Powell, M. B. [1967], *An upper bound for the chromatic number of a graph and its application to timetabling problems*, The Computer Journal, 10(1), 85-86.