# A linear algorithm for minimum dominator colorings of orientations of paths

## Michael Cary

Division of Resource Economics and Management, West Virginia University, Morgantown, WV, USA.;
macary@mix.wvu.edu

**Abstract:** In this paper we present an algorithm for finding a minimum dominator coloring of orientations of paths. To date this is the first algorithm for dominator colorings of digraphs in any capacity. We prove that the algorithm always provides a minimum dominator coloring of an oriented path and show that it runs in $\mathcal{O}(n)$ time. The algorithm is available at https://github.com/cat-astrophic/MDC-orientations_of_paths/.

**Keywords:** Dominator coloring, digraph, domination, algorithm.

## 1. Introduction and preliminaries

**L**et $G = (V, E)$ be a graph. A set $S \subset V$ is called a dominating set if every vertex of $V$ is either in $S$ or adjacent to at least one member of $S$. The domination number of a graph, $\gamma(G)$, is the size of a smallest dominating set of $G$. A dominator coloring of a graph is a proper vertex coloring of the graph which additionally satisfies the property that every vertex dominates some color class in the dominator coloring. Dominator colorings of graphs were first studied by Gera in [1–3]. Dominating sets and dominator colorings are useful for a myriad of problems and have been applied to studying electric power grids [4] and to sensors in networks [5].

Since the purpose of this paper is to introduce an algorithm for minimum dominator colorings of digraphs, it is important to know what algorithms exist for this problem in the undirected setting. A linear algorithm which finds the domination number of trees was introduced in [6]. Building from this result, [7] established a polynomial time algorithm that provides a minimum dominator coloring of trees. Additionally, it was proved in [8] that the dominator chromatic number, $\chi_d(G)$, cannot be found in polynomial time in general for many elementary families of graphs including bipartite and planar graphs. Related algorithms on domination, such as k-hop domination and k-distance domination in trees have been created [9,10], but due to the difficulty of producing exact dominating sets in graphs, each of these algorithms are confined to acyclic graphs.

Finding exact dominating sets in directed graphs is even more difficult. The dominator chromatic number of a directed graph was defined to be the minimum dominator chromatic number over all possible orientations of the underlying graph. The first results on dominator colorings of digraphs were the dominator chromatic number of paths and cycles [11], and that the dominator chromatic number of digraphs is that the dominator chromatic number of a tree is invariant under reversal of orientation [12]. However, the dominator chromatic number of relatively simple digraphs is still unknown, even for trees. Other interesting results established in [11] on dominator colorings of digraphs include the fact that the dominator chromatic number of a subgraph $H \subset G$ can be larger than the dominator chromatic number of $G$, as well as that $\limsup_{n \to \infty} \frac{\chi_d(D)}{\Delta(D)} = \infty$. That these results differ from virtually every other vertex coloring problem makes dominator colorings of digraphs of particular interest in graph theory. Independent domination in directed graphs was later studied in [13] who proved a series of existence and uniqueness theorems for independent dominating sets in different families of digraphs. Additionally, it was proven that while an independent dominating set in an orientation of a graph is an independent dominating set in the underlying graph, the converse is not necessarily true.

One of the major results from the literature on dominator colorings in digraphs was the following theorem from [11] which provides the minimum dominator chromatic number over all orientations of paths. In the

proof, many important structural characterizations relating orientations of paths and dominator colorings were established.

**Theorem 1.** *The minimum dominator chromatic number over all orientations of the path $P_n$ is given by*

$$\chi_d(P_n) = \begin{cases} k+2, & \text{if } n = 4k; \\ k+2, & \text{if } n = 4k+1; \\ k+3, & \text{if } n = 4k+2; \\ k+3, & \text{if } n = 4k+3, \end{cases}$$

*for $k \geq 1$ with the exception $\chi_d(P_6) = 3$.*

To conclude the introduction, an example of an oriented path of length five is presented below, and a minimum dominator coloring is provided in the caption. The reader is referred to [14] as the standard reference for results and terminology on domination in graphs.



**Figure 1.** An orientation of the path $P_5$ which has dominator chromatic number 3. The color classes of $P_5$ in a minimum dominator coloring may be given by $C_0 = \{v_2, v_4\}$, $C_1 = \{v_1, v_3\}$, and $C_2 = \{v_5\}$.

## 2. The algorithm

The purpose of this paper is to present the first algorithm which provides a minimum dominator coloring of a directed graph. In particular, we present an algorithm which provides a minimum dominator coloring of orientations of paths. After providing the algorithm, we will prove that it runs in $\mathcal{O}(n)$ time.

The algorithm works by sequentially through the vertex set, from $v_1$ through $v_n$, and colors each vertex in such a a way as to minimize the number of colors used in a proper dominator coloring. From Theorem 1 in [11] we know that all vertices with in-degree equal to zero must belong to the same color class. For this reason, the first thing the algorithm checks for is precisely this. Assuming this is not the case, another immediately guaranteed coloring results is that any vertex that is dominated by a vertex of out-degree one must be uniquely colored. Once these two cases are checked for, all that remains in an orientation of a path are vertices whose entire in-neighborhood consists of vertices with out-degree equal to two (notice that this may include vertices with out-degree zero or out-degree one).

It is easy to see that, for oriented paths, after removing all vertices with with at least one in-neighbor having out-degree one, what remains are subpaths in which every vertex has either out-degree zero or out-degree two, i.e., subpaths which have the following out-degree sequence pattern: $\{0, 2, 0, \ldots, 0, 2, 0\}$ (it is possible that one or both of the end vertices of such a subpath do not have out-degree zero, but since all of the out-degree two vertices are assigned the same color, and since such an end vertex would have out-degree two in the full path, we may ignore end vertices of these subpaths that do not have out-degree zero). From Theorem 1 in [11] we know that such oriented paths are minimized in terms of dominator colorings precisely when the first vertex is assigned a color $C^\star$ which is used for all vertices of out-degree zero that are not uniquely colored, and when this color is assigned to every other vertex with out-degree zero (every fourth vertex in the path). For convenience, we refer to these paths as 2-chains since they are maximal subpaths with respect to the density of vertices with out-degree two. Also, 2-chains of length three are an exception to this coloring scheme as they may have both vertices of out-degree zero belong to the same color class. This case is addressed in the algorithm as well, but first we describe the process for handling all other 2-chains.

By paying attention to where we are within a 2-chain in an oriented path, we can proceed in order through the vertex set of an oriented path and provide a minimum dominator coloring. The method used in the algorithm is to have a variable $\alpha \in \mathbb{Z}_2$ indicate whether or not we should use the color $C^\star$ or a new color

when coloring vertices of out-degree zero. The variable $\beta$ indicates whether we have established the color $C^\star$ yet. Since it turns out that 2-chains of length three may use different colors for the two vertices of out-degree zero if one vertex is colored with $C^\star$ and $C^\star$ is present elsewhere in the path (this will be proven in the theorem below), the algorithm handles 2-chains of length three by coloring both vertices the same if and only if the color $C^\star$ has not yet been established (i.e., if $\beta$ still has a value of 0).

---

**Algorithm 1** Minimum dominator coloring algorithm for oriented paths

---

1: **input** An orientated path $P_n$ of length $n$
2:
3: **initialize** $\mathcal{C} \leftarrow \{C_0\}$
4:
5: **initialize** $\mathcal{F} \leftarrow \varnothing$
6:
7: **initialize** $\alpha \leftarrow 0$
8:
9: **initialize** $\beta \leftarrow 0$
10:
11: **for** $v_i \in V(P_n)$ **do**
12:
13:     **if** $d^-(v_i) = 0$ **then**
14:
15:         Color $v_i$ with $C_0$
16:
17:         $\mathcal{F} \leftarrow \{\mathcal{F} \cup C_0\}$
18:
19:     **else if** $\exists\, v_j \in N^-(v_i)$ s.t. $d^+(v_j) = 1$ **then**
20:
21:         Color $v_i$ uniquely with a new color $C_{|\mathcal{C}|}$
22:
23:         $\mathcal{C} \leftarrow \{\mathcal{C} \cup C_{|\mathcal{C}|}\}$
24:
25:         $\mathcal{F} \leftarrow \{\mathcal{F} \cup C_{|\mathcal{C}|}\}$
26:
27:         $\alpha \leftarrow 0$                                            ▷ This indicates the end of a 2-chain
28:
29:     **else if** $\alpha = 0$ **then**
30:
31:         **if** $\beta = 0$ **then**
32:
33:             Define new color $C^\star = C_{|\mathcal{C}|}$
34:
35:             Color $v_i$ with color $C^\star$
36:
37:             $\mathcal{C} \leftarrow \{\mathcal{C} \cup C^\star\}$
38:
39:             $\mathcal{F} \leftarrow \{\mathcal{F} \cup C^\star\}$
40:
41:             **if** $n = 6$ **then**
42:
43:                 $\alpha \leftarrow 0$                      ▷ This indicates a 2-chain of length 3 or a $P_6$
44:
45:             **else**
46:
47:                 $\alpha \leftarrow 1$
48:
49:             **end if**
50:
51:             $\beta \leftarrow 1$
52:
53:         **else**
54:
55:             Color $v_i$ with existing color $C^\star$
56:
57:             $\mathcal{F} \leftarrow \{\mathcal{F} \cup C^\star\}$
58:
59:             $\alpha \leftarrow 1$
60:
61:         **end if**
62:
63:     **else**
64:
65:         Color $v_i$ uniquely with color $C_{|\mathcal{C}|}$
66:
67:         $\mathcal{C} \leftarrow \{\mathcal{C} \cup C_{|\mathcal{C}|}\}$
68:
69:         $\mathcal{F} \leftarrow \{\mathcal{F} \cup C_{|\mathcal{C}|}\}$
70:
71:         $\alpha \leftarrow 0$
72:
73:     **end if**
74:
75: **end for**
76:
77: **return** $|\mathcal{C}|$                                   ▷ The Dominator Chromatic Number is: $\chi_d(P_n) = |\mathcal{C}|$
78:
79: **return** $\mathcal{F}$                          ▷ The coloring of $V(P_n)$, i.e., $\mathcal{F}_i = c(v_i)\ \forall\, v_i \in V(P_n)$
80:

---

     Lastly we address the exception of the path $P_6$ which has either the out-degree sequence $\{0, 2, 0, 2, 0, 1\}$ or its reversal. As is stands, the algorithm would minimally color the reversal of this particular orientation of $P_6$ but not this particular orientation of $P_6$. Because this was the only exception listed in Theorem 1 from [11],

and because checking an input ($n$) will not alter the time complexity of the algorithm, we simply check for this occurrence in the same **if** statement as when we check to see if a 2-chain of length three exists.

Finally we are ready to present the algorithm (Algorithm 1). After stating the algorithm, we provide its time complexity and prove that it always provides a minimum dominator coloring of an oriented path.

Now we show that the time complexity of the algorithm is $\mathcal{O}(n)$. This result follows since we may count a constant number of things the algorithm needs to check for each vertex, hence the algorithm takes at most $cn$ steps to complete for some $c \in \mathbb{N}$.

Next we show that the algorithm provides a minimum dominator coloring of any oriented path.

**Theorem 2.** *The Minimum Dominator Coloring Algorithm 1 results in a minimum dominator coloring of every oriented path.*

**Proof.** Since any vertex with in-degree equal to zero is assigned to the same color class by this algorithm, no counterexample can come from these vertices. Since every vertex with an in-neighbor of out-degree one is colored uniquely by this algorithm, no counterexample can come from these vertices either. Thus any possible counterexample must come from the set of vertices whose entire in-neighborhood consists of vertices of out-degree two.

Since the algorithm colors each 2-chain optimally, it suffices to show that if each 2-chain is colored optimally, with the possible exception of 2-chains of length three whenever $C^\star$ already has been used, and the color $C^\star$ is common to all 2-chains which have non-uniquely colored vertices, all vertices of in-degree zero are colored similarly, and all vertices with an in-neighbor of out-degree one are colored uniquely (i.e., the conditions of the first paragraph of this proof are met), that the dominator coloring is minimum.

Clearly the color $C^\star$ must be common to all 2-chains (with non-uniquely colored vertices), else there are at least two color classes that can be combined. The MDC algorithm for orientations of paths does ensure that the color class $C^\star$ is the unique color class shared by 2-chains. With this established, it follows that each 2-chain must be (and in fact is) minimally dominator colored, except for possibly some 2-chains of length three which use the color $C^\star$ which was established prior to the instance of that 2-chain, as the only remaining vertices in our path are those vertices of 2-chains which must be colored uniquely. That the resulting dominator coloring is minimum immediately follows. □

## 3. Concluding remarks

In this paper we established the first algorithm which provides a minimum dominator coloring of directed graphs. Specifically, this algorithm provides a minimum dominator coloring for orientations of paths. We proved that this algorithm always results in a minimum dominator coloring of an oriented path, as well as that the algorithm runs in $\mathcal{O}(n)$ time.

The most likely extensions of this algorithm are to orientations of trees and cycles. While results on the dominator chromatic number of orientations of trees exist, specific results for this class of digraphs are not as established as they are for orientations of cycles, a class of graphs for which the dominator chromatic number is entirely determined. However, the acyclic structure of trees (and theoretically of directed acyclic graphs as well) lend them to being possible candidates for easy extensions of this algorithm.

To conclude this paper we mention an application of this result. For those interested, a python script, which uses this algorithm and provides a visual of a user selected orientation of a path, can be found online at the repository https://github.com/cat-astrophic/MDC-orientations_of_paths/.

**Conflicts of Interest:** "The author declares no conflict of interest."

## References

[1] Gera, R. (2007). On dominator colorings in graphs. *Graph Theory Notes of New York, 52*, 25-30.

[2] Gera, R. (2007, April). On the dominator colorings in bipartite graphs. In *Fourth International Conference on Information Technology (ITNG'07)* (pp. 947-952). IEEE.

[3] Gera, R., Rasmussen, C. W., & Horton, S. (2006). Dominator colorings and safe clique partitions. *Congress. Num. 181*, 19Ű32

*Eng. Appl. Sci. Lett.* **2020**, *3(3)*, 10-14

14

[4]   Haynes, T. W., Hedetniemi, S. M., Hedetniemi, S. T., & Henning, M. A. (2002). Domination in graphs applied to electric power networks. *SIAM Journal on Discrete Mathematics, 15*(4), 519-529.

[5]   Blair, J., Gera, R., & Horton, S. (2011). Movable dominating sensor sets in networks. *JCMCC-Journal of Combinatorial Mathematicsand Combinatorial Computing, 77*, 103.

[6]   Cockayne, E., Goodman, S., & Hedetniemi, S. (1975). A linear algorithm for the domination number of a tree. *Information Processing Letters, 4*(2), 41-44.

[7]   Merouane, H. B., & Chellali, M. (2015). An algorithm for the dominator chromatic number of a tree. *Journal of Combinatorial Optimization, 30*(1), 27-33.

[8]   Arumugam, S., Chandrasekar, K. R., Misra, N., Philip, G., & Saurabh, S. (2011, July). Algorithmic aspects of dominator colorings in graphs. *In International Workshop on Combinatorial Algorithms* (pp. 19-30). Springer, Berlin, Heidelberg.

[9]   Kundu, S. (2018). A generalized linear time algorithm for an optimal k-distance dominating set of a weighted tree. *Information Processing Letters, 130*, 58-62.

[10]  Kundu, S., & Majumder, S. (2016). A linear time algorithm for optimal k-hop dominating set of a tree. *Information Processing Letters, 116*(2), 197-202.

[11]  Cary, M. (2019). Dominator colorings of digraphs. *arXiv preprint arXiv:1902.07241.*

[12]  Cary, M., Cary, J., & Prabhu, S. (2019). Independent Domination in Directed Graphs. *arXiv preprint arXiv:1909.05121.*

[13]  Cary, M. (2019). Dominator chromatic numbers of orientations of trees. *arXiv preprint arXiv:1904.06293.*

[14]  Haynes, T. W., Hedetniemi, S., & Slater, P. (1998). *Fundamentals of domination in graphs.* CRC press.