Proceedings of the International MultiConference of Engineers and Computer Scientists 2010 Vol III,
IMECS 2010, March 17 - 19, 2010, Hong Kong

Synchronizing Coloring Of A Directed Graph

N. Karimi *

Abstract—**A synchronizing of a deterministic automaton is a word in the alphabet of colors (considered as letters) of its edges that maps the automaton to a single state. A coloring of edges of a directed graph is synchronizing if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.**

The road coloring problem is synchronizing coloring of a directed finite strongly connected graph with constant out-degree of all its vertices if the greatest common divisor of lengths of all its cycles is one. The problem was posed by Adler, Goodwin and Weiss over 30 years ago and evoked noticeable interest among the specialists in the theory of graphs, deterministic automata and symbolic dynamics.

After so many people challenge to solve this problem finally, A.N Trahtman could present a positive solution. Therefore we know that we can make a synchronizing coloring for every graph witch has the necessary conditions. In this note we are going to explain a method for appointing a synchronizing coloring and a synchronizing word, in some special cases.

1 Introduction

Let us call a directed graph $G = (V, E)$ admissible (k-admissible to be precise) if all vertices have the same out-degree k. A deterministic finite automaton (D F A) without initial and final states is obtained if we color the edges of a k-admissible directed graph with k colors in such a way that all k edges leaving any node have distinct colors. Let $\sum = \{1, 2, ..., k\}$ be the labeling alphabet. We use the standard notation $\sum^*$ for the set of words over $\sum$. Every word $w \in \sum^*$ defines a state transition function $f_w : V \to V$ on the vertex set V. The vertex $f_w(v)$ is the end point of the unique path that starts at v and whose labels read w. For a set $S \subseteq V$ we define $f_w(S) = \{f_w(v)|v \in S\}$ and $f_w^{-1}(S) = \{v \mid f_w(v) \in S\}$. Word w is called synchronizing if $f_w(V)$ is a singleton set and the automaton is called synchronized.

We have presented a method for making a synchronized automaton from a directed graph witch has necessary conditions (aperiodic and strongly connected). Note that from then on we consider just 2-admissible graph.

*Faculty Member Of Islamic Azad University Farahan Branch, Email: karimi@Iau-farahan.ac.ir

2 preliminaries

Definition 2.1. *Let $G = (V, E)$ be a directed gragh. G is said to be strongly connected if there should be a path in G between any two vertices.*

Definition 2.2. *Let $G = (V, E)$ be a directed gragh. G is said to be aperiodic if V can not be partitioned into sets $V_1, V_2, \ldots, V_d = V_0 (d > 1)$ in manner that for all edges (u, v) if $u \in V_i$ then $v \in V_{i+1}$.*

It is easy to show that the graph is aperiodic if and only if the *gcd* of the lengths of its cycles is one.

Definition 2.3. *Let $G = (V, E)$ be a 2-admissible directed gragh. Let*

$$\begin{aligned} \phi : E(G) &\longrightarrow \{r, b\} \\ uv &\longmapsto \phi(uv) \end{aligned}$$

as for every $u \in V(G)$ if uv' and uv are the only two out edges of u, then $\phi(uv) \neq \phi(uv')$. Then we say (G, ϕ) is a colored graph.

Let (G, ϕ) be a colored graph. Let $\sum = \{r, b\}$ be the labeling alphabet. We use the standard notation $\sum^*$ for the set of words over $\sum$. Every word $w \in \sum^*$ defines a state transition function $f_w : V \longrightarrow V$ on the vertex set V; the vertex $f_w(v)$ is the end point of the unique path starting at v and whose labels read w. For a set $S \subseteq V$ we define $f_w(S) = \{f_w(v) \mid v \in S\}$ and $f_w^{-1}(S) = \{v \mid f_w(v) \in S\}$.

Definition 2.4. *Word w is called synchronizing if $f_w(V)$ is a singleton set, and the automaton is called synchronized if a synchronizing word exists. A coloring of an 2-admissible directed graph is synchronized if the corresponding automaton is synchronized.*

Definition 2.5. *If word w is a synchronizing such that $f_w(V) = \{v\}$ then we call v a final vertex.*

Definition 2.6. *The matrix $A = [a_{ij}]_{(n+1)\times(n+1)}$ is said to be an adjacency matrix of the graph G if*

$$a_{ij} = \begin{cases} 1 & v_i v_j \in E(G) \\ 0 & \text{otherwise} \end{cases}$$

where $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, n$ and $V = \{v_0, v_1, \ldots, v_n\}$.

Let A be an adjacency matrix G, and (G, ϕ) is a colored graph. We define coloring matrices $R = [r_{ij}]$ and $B = [b_{ij}]$ as follow:

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

$$r_{ij} = \begin{cases} 1 & a_{ij} = 1 \quad \phi(v_i v_j) = r \\ 0 & o.w. \end{cases}$$

$$b_{ij} = \begin{cases} 1 & a_{ij} = 1 \quad \phi(v_i v_j) = b \\ 0 & o.w. \end{cases}$$

where $i = 0, 1, \ldots, n$ and $j = 0, 1, \ldots, n$. It is clear that $A = B + R$.

The road coloring conjecture: if G is an aperiodic and strongly connected admissible directed gragh, then it has synchronized coloring.

3 Coloring of G

In this section we have described our method for coloring of G. Before starting to describe our method, it would be better to explain the main idea which leads to this method.

Let G be a strongly connected and aperiodic 2-admissible directed gragh and $V = \{v_o, v_1, \ldots, v_n\}$.

Let G have synchronizing word with final vertex v_0 and $w = w_1 w_2 \ldots w_f$. So, $f_w(v_0) = v_0, f_w(v_1) = v_0, \ldots, f_w(v_n) = v_0$.

Definition 3.1. *Let $T \subseteq V$. Assume that the sets of in-edges and in-neighbors of T are denoted as follows:*

$$E(T) = \{e \in E(G) \mid e = uv;\ u \in V, v \in T\},$$

$$I(T) = \{u \in V(G) \mid \exists v \in T;\ uv \in E(G)\}.$$

Let $T_0 = \{v_0\}$, $T_k = I(T_{k-1})$ for $k = 1, 2, \ldots, f$. And let $S_0 = E(T_0)$, $S_{k-1} = E(T_{k-1})$ for $k = 1, 2, \ldots, f$.

Figure 1 shows all the paths between every v_i to v_0 following w's letters.

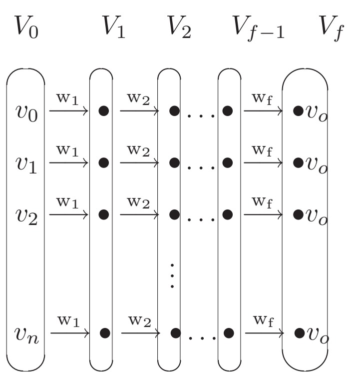

Figure 1. Word w with final vertex v_0.

Let $V_0 = V$ and $V_k = f_{w_k}(V_{k-1})$ for $k = 1, 2, \ldots, f$, and let $E_k = \{e = uv \in E(G) | u \in V_{k-1}, \phi(e) = w_k\}$ for $k = 1, 2, \ldots, f$.

It is clear that $V_k \subseteq T_{f-k}$ and $E_k \subseteq S_{f-k}$ for $k = 1, \ldots, f$.

On the other hand, all edges in E_k have the same color. Thus if we try to color G, in such a way that S_{f-k} to become mono-color as mush as possible, then, we can reach a synchronizing coloring.

How to color a graph:

Let arbitrary vertex v_0 be the final vertex. (Since G is strongly connected, the existence of synchronizing word is independent of the choice of ν_0.)

Point 3.1. *If v_0 is a vertex which has maximum in-degree, then the length of w will be shorter.*

Then, not ignoring generality, we color all of members S_0 with one of "Red" or "Blue" colors arbitrary, if any $e = (uv) \in S_0$ is colored with r (b) we will color $e' = (uv')$ with b (r). (Every vertex has 2 out degree.)

Then, we continue to color members of S_1 that haven't been colored yet. Since, at first we consider to color the edges in S_1. If blue edges are more than red edges in S_1, we will color the remaining edges in S_1 with blue and otherwise with red (as a greedy way).

Point 3.2. *If there are 2 edges in S_i for $i \geq 1$ which is leaving one common vertex, then we must not color them with a same color.*

Then, we continue to color $S_2, \ldots, S_k, \ldots$ in this way as far as all of edges are colored. In fact, we color G in such a way that all of non-colored edges between T_{k+1} and T_k become mono-colored (as a greedy way relying point 3.1).

Decomposing of A to R and B:

Now we try to decompose matrix A (adjacency matrix of G) to coloring matrices R and B according to the coloring method described at the beginning of the section. We will define matrix A' from the matrix A such that $A' = [a'_{ij}]$

$$a'_{ij} = \begin{cases} Z & a_{ij} \neq 0 \\ 0 & a_{ij} = 0 \end{cases} \quad Z = r, b,$$

then we will find R and B according to A'.

Let K be the Kth column of matrix A, that has maximum number of 1's, if all columns have the same number of 1's, we can choose any column. We change all 1's in the column K by "r".

Point 3.3. *In any row if one of the $1'$s has been colored, then the next 1 in that row should be colored opposites, at once.*

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

We will illustrate each satge by our example of matrix A,

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}.$$

We take $K = 1$ and color 1's in column by r, then color next 1's in rows 3 and 4 by b.

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ r & 0 & 0 & b & 0 \\ r & 0 & 0 & 0 & b \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

Figure 2. Coloring of rows 3 and 4 of matrix A.

Then we consider the rows of A which have been colored in the previous stage (rows 3, 4 in fig. 2) and continue to color the columns of the same number (columns 3, 4 in fig. 2). Firstly, we will check two below steps. In this step if there is no 1 in matrix A we will stop coloring. Otherwise, there will be at least a non colored 1 in these columns, because G is strongly connected.

step A: We start to change all of remained 1's with r as much as possible. (Note that when we change every 1 entry in a row we must change the other 1 in that row with opposite color at once.) Then let t_1 be the difference between the number of r's and b's in these columns.

step B: We start to change all of remained 1's with b as much as possible. Then let t_2 be the difference between the number of r's and b's in these columns.

(In figure 3 we have colored matrix A in figure 2 according steps A and B.)

$$C_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & b & r & 0 & 0 \\ r & 0 & 0 & b & 0 \\ r & 0 & 0 & 0 & b \\ 0 & 0 & r & b & 0 \end{bmatrix} \qquad D_1 = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & r & b & 0 & 0 \\ r & 0 & 0 & b & 0 \\ r & 0 & 0 & 0 & b \\ 0 & 0 & b & r & 0 \end{bmatrix}$$

step A $(t_1 = 0)$ **step B** $(t_2 = 2)$

Figure 3. We color columns 3 and 4 in fig. 2 accordoing steps A and B and let t_1 and t_2 be the difference between the numbers of rs and b's in these columns respectively.

Now if $t_1 < t_2$, then we continue coloring with step B. Otherwise, we continue with step A.

If there exists any 1 in matrix A we continue coloring by this process, frist we consider non zero rows of previous colored columns (rows 2, 3, 5 of columns 3, 4 in figure 3), then we colore the new columns which their numbers are the same as the numbers of the deretmined rows in frist step (columns 2, 3, 5). we color these new columns based on steps A and B. (See figure 4.)

$$C_2 = \begin{bmatrix} 0 & r & 0 & 0 & b \\ 0 & r & b & 0 & 0 \\ r & 0 & 0 & b & 0 \\ r & 0 & 0 & 0 & b \\ 0 & 0 & b & r & 0 \end{bmatrix} \qquad D_2 = \begin{bmatrix} 0 & b & 0 & 0 & r \\ 0 & r & b & 0 & 0 \\ r & 0 & 0 & b & 0 \\ r & 0 & 0 & 0 & b \\ 0 & 0 & b & r & 0 \end{bmatrix}$$

step A $(t_1 = 2)$ **step B** $(t_2 = 2)$

Figure 4. Coloring of $A = D_1$ according steps A and B.

If there is no 1 in matrix A we are in the final stage and we stop. But if in the final stage $t_1 = t_2$ we select coloring matrix A' as follows: Let

r_{D_f} = number of columns including r in matrix D_f,

b_{D_f} = number of columns including b in matrix D_f,

r_{C_f} = number of columns including r in matrix C_f,

b_{C_f} = number of columns including b in matrix C_f,

and $m = \min\{r_{D_f},\ r_{C_f},\ b_{D_f},\ b_{C_f}\}$.

If $m = r_C$ or $m = b_C$ then $A' = C_f$, otherwise $A' = D_f$. Then we will get matrix R and B based on A'.

In our example we see that $r_{D_2} = b_{D_2} = 4$, $r_{C_2} = b_{C_2} = 3$ and $m = \min\{r_{D_2},\ r_{C_2},\ b_{D_2},\ b_{C_2}\} = r_{D_2} = r_{C_2}$. Therefore $A' = C_2$. (See figure 5.)

$$R = \begin{bmatrix} 0 & r & 0 & 0 & 0 \\ 0 & r & 0 & 0 & 0 \\ r & 0 & 0 & 0 & 0 \\ r & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r & 0 \end{bmatrix} \qquad B = \begin{bmatrix} 0 & 0 & 0 & 0 & b \\ 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & b & 0 \\ 0 & 0 & 0 & 0 & b \\ 0 & 0 & b & 0 & 0 \end{bmatrix}$$

$$A' = C_2 = R + B$$

Figure 5. Decomposing of coloring matrix A' to two coloring matrices R and B.

4 Constructing synchronizing word

We construct synchronizing word as follows: Let $V_0 = \{v_0, v_1, \ldots, v_n\} = V$; we define subsets $V_1, V_2, \ldots$ from V_0, inductively.

$$V'_{r_k} := \phi_r(V_{k-1}) \qquad V'_{b_k} = \phi_b(V_{k-1})$$
$$|V'_{b_k}| = S'_{b_k} \qquad |V'_{r_k}| = S'_{r_k}$$

$$V_k := \begin{cases} V'_{r_k} & S'_{r_k} \leq S'_{b_k}, V'_{r_k} \neq V'_{r_j}, V'_{r_k} \neq V'_{b_j} \\ & (j = 1, 2, \ldots, k-1) \\ V'_{b_k} & o.w \end{cases}$$

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

If there exist some $f \in N$ such that:

$$|V'_{r_f}| = 1 \text{ or } |V'_{b_f}| = 1,$$

and

$$|V'_{r_i}| \neq 1 \text{ and } |V'_{b_i}| \neq 1 \text{ for all } i < f,$$

then $w = w_1 w_2 \dots w_f$, where

$$w_k = \begin{cases} r & V_k = V'_{rk} \\ b & V_k = V'_{bk} \end{cases}$$

w is the synchronizing word.

If we want to describe our method based on matrix product, indeed, we want to find a sequence of R's and B's matrices such that products of them is a matrix that has one column with all entries 1. If w is synchronizing word then the matrix $W = W_1 W_2 \dots W_f$ in which

$$W_i := \begin{cases} [r_{ij}] & w_i = r \\ [b_{ij}] & w_i = b \end{cases} \quad (i = 1, 2, \dots, f),$$

and by replacing r's and b's with 1. This is easy to show that W has just one column with all entries 1.

We have constructed an algorithm based on this way to find synchronizing word and have written its program by Q basic language. We have verified many various examples with our algorithm . We have written this program based on the matrix products R's or B's matrices. There are examples in which our algorithms are not finishing. In other word, for every $k = 1, 2, \dots$ $|V'_{r_k}| \neq 1$ and $|V'_{b_k}| \neq 1$. So, we will have two cases in hand:
A: We will run into a loop of a proper subset of V.
B: We will run into a loop of all vertices of V.

We guess that in case A, G is not strongly connected and that in case B, G is periodic.

5 Examples

In the following examples A is an adjacency matrix. We presented our examples in different shapes, but all shapes are based on our method:

Example 1 : $A = \begin{bmatrix} 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$

Figure [2, 3, 4, 5]$\Longrightarrow$

$$R = \begin{bmatrix} 0 & r & 0 & 0 & 0 \\ 0 & r & 0 & 0 & 0 \\ r & 0 & 0 & 0 & 0 \\ r & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & r & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 & 0 & b \\ 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & b & 0 \\ 0 & 0 & 0 & 0 & b \\ 0 & 0 & b & 0 & 0 \end{bmatrix}$$

$$\downarrow \begin{pmatrix} v_0 & v_1 & v_2 & v_3 & v_4 \\ v_1 & v_1 & v_0 & v_0 & v_3 \end{pmatrix} \begin{pmatrix} v_0 & v_1 & v_3 \\ v_1 & v_1 & v_0 \end{pmatrix} \begin{pmatrix} v_0 & v_1 \\ v_1 & v_1 \end{pmatrix}$$

$\Longrightarrow$ Synchronizing word is: "rrr". Note that there are some other words such as "brb".

Example 2 : $A = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \end{bmatrix}$ Starts from the second column,

$$R = \begin{bmatrix} 0 & 0 & 0 & 0 & r \\ 0 & r & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & r \\ 0 & r & 0 & 0 & 0 \\ 0 & r & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & b & 0 \\ b & 0 & 0 & 0 & 0 \\ 0 & 0 & b & 0 & 0 \\ 0 & 0 & 0 & b & 0 \end{bmatrix}$$

step 1: $\phi_r(\{v_0, v_1, v_2, v_3, v_4\}) = \{v_1, v_4\} = C_1$
$\phi_b(\{v_0, v_1, v_2, v_3, v_4\}) = \{v_0, v_2, v_3\} = D_1$
step 2: $\phi_r(C_1) = \{v_1\}$ $\quad \phi_b(C_1) = \{v_3\}$

$\Longrightarrow$ Synchronizing words are: "rr" with final vertex: v_1, and "rb" with final vertex: v_3.

Example 3 : $A = \begin{bmatrix} 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 \end{bmatrix}$ Starts from the third column

$$R = \begin{bmatrix} 0 & 0 & 0 & r & 0 \\ 0 & 0 & r & 0 & 0 \\ r & 0 & 0 & 0 & 0 \\ 0 & 0 & r & 0 & 0 \\ 0 & 0 & r & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & b & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b \\ 0 & 0 & 0 & b & 0 \\ b & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & b \end{bmatrix}$$

$$\downarrow \begin{pmatrix} v_0 & v_1 & v_2 & v_3 & v_4 \\ v_3 & v_2 & v_0 & v_2 & v_2 \end{pmatrix} \begin{pmatrix} v_0 & v_2 & v_3 \\ v_1 & v_3 & v_0 \end{pmatrix} \begin{pmatrix} v_0 & v_1 & v_3 \\ v_3 & v_2 & v_2 \end{pmatrix}$$
$$\begin{pmatrix} v_2 & v_3 \\ v_0 & v_2 \end{pmatrix} \begin{pmatrix} v_0 & v_2 \\ v_3 & v_0 \end{pmatrix} \begin{pmatrix} v_0 & v_3 \\ v_1 & v_0 \end{pmatrix} \begin{pmatrix} v_0 & v_1 \\ v_1 & v_4 \end{pmatrix} \begin{pmatrix} v_1 & v_4 \\ v_2 & v_2 \end{pmatrix}$$

$\Longrightarrow$ Synchronizing word is: "rbr^3b^2r".

ISBN: 978-988-18210-5-8
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Eample 4 : $A = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 0 \end{bmatrix}$

$$R = \begin{bmatrix} 0 & r & 0 & 0 \\ r & 0 & 0 & 0 \\ 0 & r & 0 & 0 \\ r & 0 & 0 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 & 0 & 0 & b \\ 0 & 0 & b & 0 \\ 0 & 0 & 0 & b \\ 0 & 0 & b & 0 \end{bmatrix}$$

$\downarrow \begin{pmatrix} v_0 & v_1 & v_2 & v_3 \\ v_1 & v_0 & v_1 & v_0 \end{pmatrix} \begin{pmatrix} v_0 & v_1 \\ v_3 & v_2 \end{pmatrix} \begin{pmatrix} v_2 & v_3 \\ v_1 & v_0 \end{pmatrix}$

$\Longrightarrow$ We run into a loop of all vertices of V; and G is periodic.

Acknowledgement

We express our warm thanks with Professor B. Nashvadiyan for his helps to us in editing.

References

[1] R. L. Adler, L.W. Goodwyn, and B. Weiss., *Equivalence of topological Markov shifts*, Israel j. Math. $27 : 49 - 63, (1977)$.

[2] G.L. O'Brien., *The road-coloring problem*, Israel j. Math. $39 : 145 - 154, (1981)$

[3] J. Friedman., *On the road coloring*, Proc. Amer. Math. Soc. $110 : 1133 - 1135, (1990)$.

[4] J. Kari., *Synchronizing finite automata on Eulerian digragh*, Theoret. comput. sci, $295(1 - 3) : 223 - 232, (2003)$.

[5] R.L. Adler and B. Weiss., *Similarity of automorphisms of the torus*, Memooirs of the American mathemathical society, providence, RI, 1970.

[6] A.N Trahtman., *The toad coloring and cerny conjecture*, proc.prague stringology conferece $2008, 1 - 12$.