A Subquadratic Algorithm for Road Coloring

A.N. Trahtman* **

Bar-Ilan University, Dep. of Math., 52900, Ramat Gan, Israel

Abstract. The synchronizing word 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 the problem of synchronizing coloring of a directed finite strongly connected graph with constant outdegree of all its vertices if the greatest common divisor of lengths of all its cycles is one. The problem was posed by Adler, Goodwyn and Weiss in 1970 and had evoked many years a noticeable interest among the specialists in the theory of graphs, deterministic automata and symbolic dynamics.

A polynomial time algorithm of $O(n^3)$ complexity in the most worst case and quadratic in most cases for the road coloring of the considered graph is presented below. The work is based on my recent positive solution of the road coloring problem.

Keywords: algorithm, road coloring, graph, deterministic finite automaton, synchronization.

Introduction

The road coloring problem originates in [2] and was stated explicitly in [1] for a strongly connected directed finite graph with constant outdegree of all its vertices where the greatest common divisor (gcd) of lengths of all its cycles is one. The edges of the graph are unlabelled. The task is to find a labelling of the edges that turns the graph into a deterministic finite automaton possessing a synchronizing word. So the road coloring problem is connected with the problem of existence of synchronizing word for deterministic complete finite automaton.

The condition on gcd is necessary [1], [6]. It can be replaced by the equivalent property that there does not exist a partition of the set of vertices on subsets $V_1, V_2, \ldots, V_k = V_1$ ($k > 1$) such that every edge which begins in $V_i$ has its end in $V_{i+1}$ [6]. The outdegree of the vertex can be considered also as the size of an alphabet where the letters denote colors.

The road coloring problem is important in automata theory: a synchronizing coloring makes the behavior of an automaton resistant against input errors since,

* Email: trakht@macs.biu.ac.il

** http://www.cs.biu.ac.il/~trakht
after detection of an error, a synchronizing word can reset the automaton back to its original state, as if no error had occurred. The problem appeared first in the context of symbolic dynamics and is important also in this area. Together with the Černy conjecture [17], [19], the road coloring problem belonged to the most fascinating problems in the theory of finite automata. The problem is discussed even in "Wikipedia" - the popular Internet Encyclopedia. However, at the same time it was considered as a "notorious open problem" [13] and "unfeasible" [9].

For some results in this area see [4], [5], [7], [8], [11], [12], [15], [16], DNA algorithm approach see in [10]; a detailed history of investigations can be found in [5]. The final positive solution of the problem stated in [21].

An algorithm for road coloring oriented on DNA computing [10] is based on massive parallel computing of sequences of length $O(n^3)$. The implementation of the algorithm as well as the implementation of effective DNA computing is still a problem of future.

We present a polynomial time algorithm for the road coloring of the considered graph. The algorithm is based on the recent solution of the problem [21]. Some results from there are strengthened or presented in another wording. The concept from [7] of the weight of a vertex supposed by Friedman and the concept of a stable pair of states of Culik, Karhumaki and Kari [6], [12] with corresponding results and consequences are also used below.

The time and space complexity of the algorithm for graph with $n$ vertices and $d$ outgoing edges of any vertex is $O(n^3d)$ in the most worst case and quadratic or less in the majority of cases.

**Preliminaries**

A finite directed strongly connected graph with constant outdegree of all its vertices where the gcd of lengths of all its cycles is one will be called AGW graph as aroused by Adler, Goodwyn and Weiss.

$|P|$ - the size of the subset $P$ of states of an automaton (of vertices of a graph). If there exists a path in an automaton from the state $p$ to the state $q$ and the edges of the path are consecutively labelled by $\sigma_1, ..., \sigma_k$, then for $s = \sigma_1...\sigma_k \in \Sigma^+$ let us write $q = ps$.

Let $Ps$ be the map of the subset $P$ of states of an automaton by help of $s \in \Sigma^+$ and let $Ps^{-1}$ be the maximal set of states $Q$ such that $Qs \subseteq P$. For the transition graph $\Gamma$ of an automaton let $\Gamma s$ denote the map of the set of states of the automaton.

A word $s \in \Sigma^+$ is called a synchronizing word of the automaton with transition graph $\Gamma$ if $|\Gamma s| = 1$.

A coloring of a directed finite graph is synchronizing if the coloring turns the graph into a deterministic finite automaton possessing a synchronizing word.

A pair of distinct states $p, q$ of an automaton (of vertices of the transition graph) will be called synchronizing if $ps = qs$ for some $s \in \Sigma^+$. In the opposite case, if for any $s \neq ps \neq qs$, we call the pair deadlock.
A synchronizing pair of states \( p, q \) of an automaton is called stable if for any word \( u \) the pair \( pu, qu \) is also synchronizing [6], [12].

We call the set of all outgoing edges of a vertex a bunch if all these edges are incoming edges of only one vertex.

Let \( u \) be a left eigenvector with positive components having no common divisor of adjacency matrix of a graph with vertices \( p_1, ..., p_n \). The i-th component \( u_i \) of the vector \( u \) is called the weight of the vertex \( p_i \) and denoted by \( w(p_i) \). The sum of the weights of the vertices from a set \( D \) is denoted by \( w(D) \) and is called the weight of \( D \) [7].

The subset \( D \) of states of an automaton (of vertices of the transition graph \( \Gamma \) of the automaton) such that \( w(D) \) is maximal and \( |Ds| = 1 \) for some word \( s \in \Sigma^+ \) let us call \( F \)-maximal as introduced by Friedman [7].

The subset \( \Gamma s \) of states (of vertices of the transition graph \( \Gamma \)) for some word \( s \) such that every pair of states from the set is deadlock will be called an \( F \)-clique.

1 Some properties of \( F \)-clique and of coloring free of stable pairs

The road coloring problem was formulated for AGW graphs [1] and only such graphs are considered below. Let us formulate two useful results from [7] and [12] in the following form:

**Theorem 1** [7] There exists a partition of \( \Gamma \) on \( F \)-maximal sets (of the same weight).

**Theorem 2** [12] Let us consider a coloring of AGW graph \( \Gamma \). Let \( \rho \) be a binary relation on the set of states of the obtained automaton; suppose \( p \rho q \) if the pair of states \( p, q \) is stable.

Then \( \rho \) is a congruence relation, \( \Gamma / \rho \) presents an AGW graph and synchronizing coloring of \( \Gamma / \rho \) implies synchronizing recoloring of \( \Gamma \).

**Lemma 1** [21] Let \( w \) be the weight of \( F \)-maximal set of the AGW graph \( \Gamma \) via some coloring. Then the size of every \( F \)-clique of the coloring is the same and equal to \( w(\Gamma)/w \) (the size of partition of \( \Gamma \) on \( F \)-maximal sets).

Proof. Two states from an \( F \)-clique could not belong to one \( F \)-maximal set because this pair is not synchronizing. By Theorem 1 there exists a partition of \( \Gamma \) on \( F \)-maximal sets of weight \( w \). So the partition consists from \( w(\Gamma)/w \) \( F \)-maximal sets and to every \( F \)-maximal set belongs at most one state from \( F \)-clique. Consequently, the size of any \( F \)-clique is not greater than \( w(\Gamma)/w \).

Let \( \Gamma s \) be an \( F \)-clique. The sum of the weights \( qs^{-1} \) for all \( q \in \Gamma s \) is the weight of \( \Gamma \). So

\[
w(\Gamma) = \sum_{q \in \Gamma s} w(qs^{-1})
\]

The number of addends (the size of the \( F \)-clique) is not greater than \( w(\Gamma)/w \). The weight of the set \( qs^{-1} \) for every \( q \in \Gamma s \) is not greater than \( w \). Therefore
\(q\)s is an \(F\)-maximal set of weight \(w\) for every \(q \in \Gamma\) and the size of any \(F\)-clique is \(w(\Gamma)/w\), the number of \(F\)-maximal sets in the corresponding partition of \(\Gamma\).

**Lemma 2** \[21\] Let \(F\) be \(F\)-clique via some coloring of AGW graph \(\Gamma\). For any word \(s\) the set \(Fs\) is also an \(F\)-clique and any state [vertex] \(p\) belongs to some \(F\)-clique.

Proof. Any pair \(p, q\) from an \(F\)-clique \(F\) is a deadlock, therefore the pair \(ps, qs\) from \(Fs\) also is a deadlock and therefore \(Fs\) is an \(F\)-clique.

For \(p\) from an \(F\)-clique \(F\) and arbitrary \(r\) there exists a word \(s\) such that \(r = ps\), whence \(r\) belongs to the \(F\)-clique \(Fs\).

**Lemma 3** Let \(A\) and \(B\) (\(|A| > 1\)) be distinct \(F\)-cliques via some coloring of the AGW graph \(\Gamma\) such that \(|A| - |A \cap B| = 1\). Then for \(p \in A \setminus A \cap B\) and \(q \in B \setminus A \cap B\) the pair \((p, q)\) is stable.

Proof. By Lemma \[1\] \(|A| = |B|\). So \(|B| - |A \cap B| = 1\), too. If the pair of states \(p \in A \setminus B\) and \(q \in B \setminus A\) is not stable then for some word \(s\) the pair \((ps, qs)\) is a deadlock. Any pair of states from the \(F\)-clique \(A\) and from the \(F\)-clique \(B\) as well as from \(F\)-cliques \(As\) and \(Bs\) is a deadlock. So any pair of states from the set \((A \cup B)s\) is a deadlock. One has \(|(A \cup B)s| = |As| + 1 = |A| + 1 > |A|\).

In view of Theorem \[1\] there exists a partition of size \(|A|\) (Lemma \[1\]) of \(\Gamma\) on \(F\)-maximal sets. To every \(F\)-maximal set belongs at most one state from \((A \cup B)s\) because every pair of states from this set is a deadlock and no deadlock could belong to an \(F\)-maximal set. This contradicts the fact that the size \(|A| + 1\) of the set \((A \cup B)s\) is greater than \(|A|\).

**Lemma 4** \[21\] If some vertex of AGW graph \(\Gamma\) has two incoming bunches then the beginnings of the bunches form a stable couple by any coloring.

Proof. If a vertex \(p\) has two incoming bunches from \(q\) and \(r\), then the couple \(q, r\) is stable for any coloring because \(q\sigma = r\sigma = p\) for any \(\sigma \in \Sigma\).

2 The spanning subgraph of AGW graph

**Définition 1** Let us call a subgraph \(S\) of the AGW graph \(\Gamma\) a spanning subgraph of \(\Gamma\) if to \(S\) belong all vertices of \(\Gamma\) and exactly one outgoing edge of any vertex.

A maximal subtree of the spanning subgraph \(S\) with root on a cycle from \(S\) and having no common edges with cycles from \(S\) is called a tree of \(S\).

The length of path from a vertex \(p\) through the edges of the tree of the spanning set \(S\) to the root of the tree is called the level of \(p\) in \(S\).

**Remark 1** Any spanning subgraph \(S\) consists of disjoint cycles and trees with roots on cycles; any tree and cycle of \(S\) is defined identically, the level of the vertex from cycle is zero, the vertices of trees except root have positive level, the vertex of maximal positive level has no incoming edge from \(S\). The edges of every given color by any coloring form a spanning subgraph and for any spanning subgraph there exists a corresponding coloring.
Lemma 5. Let $N$ be a set of vertices of level $n$ from some tree of the spanning subgraph $S$ of AGW graph $\Gamma$. Then via a coloring of $\Gamma$ such that all edges of $S$ have the same color $\alpha$, for any $F$-clique $F$ holds $|F \cap N| \leq 1$.

Proof. Some power of $\alpha$ synchronizes all states of given level of the tree and maps them into the root. Any couple of states from an $F$-clique could not be synchronized and therefore could not belong to $N$.

Lemma 6. Let AGW graph $\Gamma$ have a spanning subgraph $R$ of cycles (without trees). Then non-trivial graph $\Gamma$ has another spanning subgraph with exactly one vertex of maximal positive level.

Proof. The spanning subgraph $R$ has only cycles and therefore the levels of all vertices are equal to zero. In view of gcd =1 in the strongly connected graph $\Gamma$, not all edges belong to a bunch. Therefore there exist two edges $u = p \rightarrow q \notin R$ and $v = p \rightarrow s \in R$ with common first vertex $p$ but such that $q \neq s$. Let us replace the edge $v = p \rightarrow s$ from $R$ by $u$. Then only the vertex $s$ has maximal level $L > 0$ in the new spanning subgraph.

Lemma 7. Let any vertex of an AGW graph $\Gamma$ have no two incoming bunches. Let $R$ be a spanning subgraph and let its tree $T$ with the root $r$ on cycle $H$ have a vertex $p$ of maximal level. Let us consider 1) replacing of edge from $R$ by an incoming edge of $p$ with the same beginning, 2) replacing in $R$ of an incoming edge of a root on path in tree from $p$, 3) replacing in $R$ of an incoming edge of a root from $H$.

Suppose that at most two such consecutive operations on $H$ do not increase the number of edges in cycles. Then some new spanning subgraph has all vertices of maximal positive level in only one non-trivial tree.

Proof. In view of Lemma 6, suppose that $R$ has non-trivial trees. Further consideration is necessary only if at least two vertices of level $L$ belong to distinct trees of $R$ with distinct roots.

Let the edge $\bar{b} = b \rightarrow r \in T$ belong to the path from $R$ of the maximal length $L$ with beginning in $p$. Suppose $\bar{c} = c \rightarrow r \in H$. There exists also an edge $\bar{a} = a \rightarrow p$ that does not belong to $R$ because $\Gamma$ is strongly connected and $p$ has no incoming edge from $R$. Let $\bar{w} = a \rightarrow d$ belong to $R$.

Let us consider the path from $p$ to $r$ of maximal length $L$ in $T$. Our aim is to extend the maximal level of the vertex on the extension of the tree $T$ much more than the maximal level of vertex of other trees from $R$. We plan to use the three aforesaid ways. If one of the ways does not succeed let us go to the
next assuming the situation in which the previous way fails and excluding the successfully studied cases. So we diminish the considered domain. We can use sometimes two changes together. Let us begin with

1) Suppose first \(a \notin H\). If \(a\) belongs to a path in \(T\) from \(p\) to \(r\) then a new cycle with part of the path and edge \(a \to p\) is added to \(R\) extending the number of vertices in its cycles in spite of the condition of lemma. In opposite case the level of \(a\) is \(L + 1\) in the new spanning subgraph and the vertex \(r\) is a root of the new tree containing all vertices of maximal level (the vertex \(a\) or its ancestors in \(R\)). So let us assume \(a \in H\) and suppose \(\bar{w} = a \to d \in H\). In this case the vertices \(p, r\) and \(a\) belong to a cycle \(H_1\) with new edge \(\bar{a}\) of a new spanning subgraph \(R_1\). So we have the cycle \(H_1 \in R_1\) instead of \(H \in R\). If the length of path from \(r\) to \(a\) in \(H\) is \(r_1\) then \(H_1\) has length \(L + r_1 + 1\). A path to \(r\) from the vertex \(d\) of the cycle \(H\) remains in \(R_1\). Suppose its length is \(r_2\). So the length of the cycle \(H\) is \(r_1 + r_2 + 1\). The length of the cycle \(H_1\) is not greater than the length of \(H\) because by the condition of lemma the number of edges in the cycles of the spanning subgraph could not grow. So \(r_1 + r_2 + 1 \geq L + r_1 + 1\), whence \(r_2 \geq L\).

If \(r_2 > L\), then the length \(r_2\) of the path from \(d\) to \(r\) in a tree of \(R_1\) (and the level of \(d\)) is greater than \(L\) and the level of \(d\) (or of some other ancestor of \(r\) in a tree from \(R_1\)) is the desired unique maximal level.

So assume for further consideration \(L = r_2\) and \(a \in H\). Analogously, for any vertex of maximal level \(L\) with root in the cycle \(H\) and incoming edge from a vertex \(a_1\) the proof can be reduced to the case \(a_1 \in H\) and \(L = r_2\) for the corresponding value \(r_2\).

2) Suppose the set of outgoing edges of the vertex \(b\) is not a bunch. So one can replace in \(R\) the edge \(\bar{b}\) from the vertex \(b\) by an edge \(\bar{v}\) from \(b\) to a vertex \(v \neq r\). The vertex \(v\) could not belong to \(T\) because in this case a new cycle is added to \(R\), but the number of vertices in the cycles of the spanning subgraph could not grow.

If the vertex \(v\) belongs to another tree of \(R\) but not to cycle, then \(T\) is a part of a new tree \(T_1\) with a new root of a new spanning subgraph \(R_1\) and the path from \(p\) to the new root is extended. Therefore only the tree \(T_1\) has states of new maximal level.

If \(v\) belongs to some cycle \(H_2 \neq H\) from \(R\), then together with replacing \(\bar{b}\) by \(\bar{v}\), we replace also the edge \(\bar{w}\) by \(\bar{a}\). So we extend the path from \(p\) to the new root \(v\) at least by the edge \(\bar{a} = a \to p\) and by almost all edges of \(H\). Therefore the new maximal level \(L_1 > L\) has either the vertex \(d\) or its ancestors from the old spanning subgraph \(R\).

Now it remains only the case when \(v\) belongs to the cycle \(H\). The vertex \(p\) also has level \(L\) in new tree \(T_1\) with root \(v\). The only difference between \(T\) and \(T_1\) (just as between \(R\) and \(R_1\)) is the root and the incoming edge of the root. The new spanning subgraph \(R_1\) has the same number of vertices in cycles just as \(R\). Let \(r_2\) be the length of the path from \(d\) to the new root \(v \in H\).

For the spanning subgraph \(R_1\), one can obtain \(L = r_2\) just as it was done on the step 1) for \(R\). From \(v \neq r\) follows \(r_2 \neq r_2\), though \(L = r_2\) and \(L = r_2\).

So for further consideration suppose that the set of outgoing edges of the vertex
b is a bunch to r.
3) The set of outgoing edges of the vertex c is not a bunch because r has another bunch from b in virtue of the lemma condition.
Let us replace in R the edge $\tilde{c}$ by an edge $\tilde{u} = c \rightarrow u$ such that $u \neq r$. The vertex u could not belong to the tree T because one has in this case a cycle with all vertices from H and some vertices of T whence its length is greater than $|H|$ and so the number of vertices in the cycles of a new spanning subgraph grows in spite of the assumption of lemma.
If the vertex u does not belong to T, then the tree T is a part of a new tree with a new root and the path from p to the new root is extended at least by a part of H from the former root r. The new level of p therefore is maximal and greater than the level of any vertex in some another tree.
Thus anyway we obtain a spanning subgraph with vertices of maximal level in one non-trivial tree.

**Lemma 8** Let any vertex of an AGW graph $\Gamma$ have no two incoming bunches. Let R be a spanning subgraph and let its tree T with the root r on cycle H have all vertices of maximal level L and one of them is the vertex p. Let us consider
1) replacing of edge from R by an incoming edge of p with the same beginning,
2) replacing in R of an incoming edge of a root on path in tree from p,
3) replacing in R of an incoming edge of a root from H.
Suppose that at most two such operations do not increase the number of edges in cycles. Then by coloring of R by color $\alpha$ the pair $p\alpha^{L-1}$, $p\alpha^{L+|H|-1}$ is stable.

Proof. By Lemma 7 $\Gamma$ has a spanning subgraph R such that all vertices of maximal positive level L belong to one tree of R. Let us give to the edges of R the color $\alpha$ and color the remaining edges of $\Gamma$ by other colors arbitrarily.
By Lemma 2 in a strongly connected transition graph for every word s and F-clique F of size $|F| > 1$, the set $F$s also is an F-clique (of the same size by Lemma 1) and for any state p there exists an F-clique F such that $p \in F$.
In particular, some F has non-empty intersection with the set N of vertices of maximal level L. The set N belongs to one tree, whence by Lemma 3 this intersection has only one vertex. The word $\alpha^{L-1}$ maps F on an F-clique $F_1$ of size $|F|$. One has $|F_1 \setminus C| = 1$ because the sequence of edges of color $\alpha$ from any tree of R leads to the root of the tree, the root belongs to a cycle colored by $\alpha$ from C and only for the set N of vertices of maximal level holds $N \alpha^{L-1} \not\subseteq C$.
So $|N \alpha^{L-1} \cap F_1| = |F_1 \setminus C| = 1$, $p\alpha^{L-1} \in F_1 \setminus C$ and $|C \cap F_1| = |F_1| - 1$.
Let the integer $m$ be a common multiple of the lengths of all considered cycles from C colored by $\alpha$. So for any r from C as well as from $F_1 \cap C$ holds $\alpha^m = r$.
Therefore for an F-clique $F_2 = F_1 \alpha^m$ holds $F_2 \subseteq C$ and $C \cap F_1 = F_1 \cap F_2$.
Thus two F-cliques $F_1$ and $F_2$ of size $|F_1| > 1$ have $|F_1| - 1$ common vertices. So $|F_1 \setminus (F_1 \cap F_2)| = 1$, whence by Lemma 3 the pair of states $p\alpha^{L-1}$ from $F_1 \setminus (F_1 \cap F_2)$ and q from $F_2 \setminus (F_1 \cap F_2)$ is stable. Evidently that $q = p\alpha^{L+m-1}$.

**Theorem 3** Every AGW graph has synchronizing coloring.

The proof follows from Theorem 2, Lemma 4 and Lemma 8.
Theorem 4  Let every vertex of strongly connected directed graph $\Gamma$ have the same number of outgoing edges. Then $\Gamma$ has synchronizing coloring if and only if the greatest common divisor of lengths of all its cycles is one.

In view of Theorem 3 we must prove only the necessity of the condition on gcd. Proof [1], [6]. Suppose $d > 1$ is the greatest common divisor of lengths of all cycles of $\Gamma$. Let us consider a tree $T$ with root $p$ and with all vertices of the graph and define a function $t$ on the set of vertices. Suppose $t(p) = 0$ and for every edge $r \to q$ of the tree suppose $t(r) = t(q) + 1$ (modulo $d$). So $t(q) < d$ for every vertex $q$.

Let the edge $u \to v$ be outside of $T$. If $t(u) \neq t(v) + 1$ (modulo $d$) then two paths from $p$ to $v$ through the edge $u \to v$ and the edges of $T$ and through only the edges of $T$ have not equal (modulo $d$) lengths. Therefore in strongly connected graph $\Gamma$ there are two cycles having not equal lengths (modulo $d$). It contradicts to the choice of $d$. So for any edge $u \to v$ one has $t(u) = t(v) + 1$ (modulo $d$). Consequently by whatever coloring for any word $s$ of the colors one has $t(us) = t(vs) + 1$. So any word $s$ could not unite $v$ and $u$, whence $\Gamma$ has no synchronizing coloring.

Lemma 9  Let AGW graph $\Gamma$ have two cycles $C_u$ and $C_v$ either with one common vertex $p_1$ or with a common sequence $p_1, \ldots, p_k$, such that all incoming edges of $p_i$ form a bunch from $p_{i+1}$ (i < k). Let $u \in C_u$ and $v \in C_v$ be the edges of the cycles leaving $p_1$. Let $T$ be a maximal subtree of $\Gamma$ with root $p_1$ and edges from $C_u$ and $C_v$ except $u$ and $v$.

Then the subtree $T$ by adding one of the edges $u$ or $v$ turns in spanning subgraph with vertices of maximal level in one tree of the spanning subgraph.

Proof. Let us add to $T$ either $u$ or $v$ and then find the maximal levels of vertices in both cases. The vertex $p_i$, for $i > 1$ could not be the root of a tree. If any tree of spanning subgraph with vertex of maximal level has the root $p_1$ then in both opportunities the lemma holds. If some tree of spanning subgraph with vertex of maximal level has the root only on $C_u$ then let us choose the adding of $v$. In this case the level of the considered vertex is growing and only the new tree with root $p_1$ has vertices of maximal level. In the case of root on $C_v$ let us add $u$.

3 The algorithm for synchronizing coloring

Let us begin from arbitrary coloring of AGW graph $\Gamma$ with $n$ vertices and $d$ outgoing edges of any vertex. The considered $d$ colors define $d$ spanning subgraphs of the graph.

We keep images of vertices and colored edges from generic graph by any transformation and homomorphism.

If there exists a loop in $\Gamma$ then let us color the edges of a tree with root in the vertex of loop by one color. The other edges may be colored arbitrarily. The coloring is synchronizing [1].

In the case of two incoming bunches of one vertex the beginnings of these bunches
form a stable pair by any coloring (Lemma 4). We unite both vertices in the homomorphic image of the graph (Theorem 2) and obtain according the theorem a new AGW graph of the size $|\Gamma| - 1$.

The linear search of two incoming bunches and of loop can be made on any stage of the algorithm.

Find the parameters of the spanning subgraph: levels of all vertices, the number of vertices (edges) in cycles, for any vertex let us keep its tree and the cycle of the root of the tree. We form the set of vertices of maximal level and choose from the set of trees a tree $T$ with vertex $p$ of maximal level. This step is linear and used by any recoloring.

1) If there are two cycles with one common vertex then we use the Lemma 9 and find a spanning subgraph $R_1$ such that any vertex $p$ of maximal level $L$ belongs to one tree with root on a cycle $H$. Then after coloring edges of $R_1$ by color $\alpha$ we find stable pair $q = p^{L-1+|H|}$ and $s = p^{L-1}$ (Lemma 8) and go to the step 3).

2) Let us consider now the three replacements from Lemma 7 and find the number of edges in cycles and other parameters of the spanning subgraph of the given color. If the number of edges in cycles is growing, then the new spanning subgraph must be considered and the new parameters of the subgraph must be found. In opposite case, after at most $3d$ steps, by Lemma 7 there exists a tree $T_1$ with root on cycle $H_1$ of a spanning subgraph $R_1$ such that any vertex $p$ of maximal level $L$ belongs to $T_1$.

Suppose the edges of $R_1$ are colored by color $\alpha$. Then the vertices $q = p^{L-1+|H_1|}$ and $s = p^{L-1}$ by Lemma 8 form a stable pair.

3) Let us finish the coloring and find the subsequent stable pairs of the pair $(s, q)$ using appropriate coloring. Then we go to the homomorphic image $\Gamma_i / \rho$ (Theorem 2) of considered graph $\Gamma_i$ ($O(|\Gamma_i| m, d)$ complexity where $m_i$ is the size of the map $\Gamma_i$). Then we repeat the procedure with new graph $\Gamma_{i+1}$ of a smaller size. So the overall complexity of this step of the algorithm is $O(n^2 d)$ in majority of cases and $O(n^3 d)$ if the number of edges in cycles grows slowly, $m_i$ decreases also slowly, loops do not appear and the case of two ingoing bunches emerges rarely (the most worst case).

Let $\Gamma_{i+1} = \Gamma_i / \rho_{i+1}$ on some stage $i + 1$ have synchronizing coloring. For every stable pair $q, p$ of vertices from $\Gamma_i$ there exists a pair of corresponding outgoing edges that reach either another stable pair or one vertex. This pair of edges is mapped on one image edge of $\Gamma_{i+1}$. So let us give the color of the image to preimages and obtain on this way a synchronizing coloring of $\Gamma_i$. This step is linear. So the overall complexity of the algorithm is $O(n^3 d)$ in the most worst case and $O(n^2 d)$ in most cases. Sometimes it is even linear.

### 3.1 Check the necessary conditions of synchronizing coloring

The algorithm is based on the Theorem 4. One must check the existence of sink strongly connected component $S$ and check the condition on $\gcd$ in $S$.

Let us use the linear algorithm of finding of strongly connected component $(SCC)$ [3], [18]. Then we mark all SCC having outgoing edges to others SCC.
If only one SCC does not be marked then sink exists and belongs to this SCC $H$. In opposite case the synchronizing coloring does not exist.

Then let us study SCC $H$. Let $p$ be a vertex from $H$. Suppose $d(p) = 1$. For an edge $r \rightarrow q$ where $d(r)$ is already defined and $d(q)$ is not defined suppose $d(q) = d(r) + 1$. If $d(q)$ is defined let us add the difference $\text{abs}(d(q) - 1 - d(r))$ to the set $D$ and count the gcd of the integers from $D$. If $\text{gcd} = 1$ the graph has synchronizing coloring. If after checking all edges from $H$ the $\text{gcd} \neq 1$ the answer is negative.

The verifying of the necessary conditions of synchronizing coloring is linear.

References