Discrete Applied Mathematics 169 (2014) 15–29

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

A quadratic algorithm for road coloring

Marie-Pierre Béal *, Dominique Perrin

Université Paris-Est, Laboratoire d'informatique Gaspard-Monge CNRS UMR 8049, 5 boulevard Descartes, 77454 Marne-la-Vallée, France

ARTICLE INFO

Article history:
Received 30 May 2012
Received in revised form 25 November 2013
Accepted 5 December 2013
Available online 31 January 2014

Keywords:
Road Coloring Problem
Synchronized directed graphs
Synchronization of automata

ABSTRACT

The Road Coloring Theorem states that every aperiodic directed graph with constant out-degree has a synchronized coloring. This theorem had been conjectured during many years as the Road Coloring Problem before being settled by A. Trahtman. Trahtman's proof leads to an algorithm that finds a synchronized labeling with a cubic worst-case time complexity. We show a variant of his construction with a worst-case complexity which is quadratic in time and linear in space. We also extend the Road Coloring Theorem to the periodic case.

1. Introduction

Imagine a map with roads which are colored in such a way that a fixed sequence of colors, called a homing sequence, leads the traveler to a fixed place whatever the starting point is. Such a coloring of the roads is called synchronized and finding a synchronized coloring is called the Road Coloring Problem. In terms of graphs, it consists in finding a synchronized labeling in a directed graph.

The Road Coloring Theorem states that every aperiodic directed graph with constant out-degree has a synchronized coloring (a graph is aperiodic if it is strongly connected and the gcd of the length of the cycles is equal to 1). It has been conjectured under the name of the Road Coloring Problem by Adler, Goodwin, and Weiss [2], and solved for many particular types of automata (see for instance [2,23,9,19,16,25]). Trahtman settled the conjecture in [29]. In this paper, by Road Coloring Problem we understand the algorithmic problem of finding a synchronized coloring on a given graph (and not the existence of a polynomial algorithm which is solved by the Road Coloring Theorem).

Solving the Road Coloring problem in each particular case is not only a puzzle but has many applications in various areas like coding or design of computational systems. These systems are often modeled by finite-state automata (*i.e.* graphs with labels). Due to some noise, the system may take a wrong transition. This noise may for instance result from the physical properties of sensors, from unreliability of computational hardware, or from insufficient speed of the computer with respect to the arrival rate of input symbols. It turns out that the asymptotic behavior of synchronized automata is better than the behavior of unsynchronized ones (see [12]). Synchronized automata are thus less sensitive to the effect of noise.

In the domain of coding, automata with outputs (*i.e.* transducers) can be used either as encoders or as decoders. When they are synchronized, the behavior of the coder (or of the decoder) is improved in the presence of noise or errors (see [4,18]). For instance, the well-known Huffman compression scheme leads to a synchronized decoder provided the lengths of the codewords of the Huffman code are relatively prime. It is also a consequence of the Road Coloring Theorem that coding schemes for constrained channels can have sliding block decoders and synchronized encoders (see [1,21]).

* Corresponding author. Tel.: +33 1 60 95 75 67; fax: +33 1 60 95 75 57.
E-mail addresses: beal@univ-mlv.fr (M.-P. Béal), perrin@univ-mlv.fr (D. Perrin).

0166-218X/$ – see front matter
http://dx.doi.org/10.1016/j.dam.2013.12.002

Trahtman's proof is constructive and leads to an algorithm that finds a synchronized labeling with a cubic worst-case time complexity [29,31]. The algorithm consists in a sequence of flips of edges going out of some state so that the resulting automaton is synchronized. One first searches a sequence of flips leading to an automaton which has a so-called stable pair of states (*i.e.* with good synchronizing properties). One then computes the quotient of the automaton by the congruence generated by the stable pairs. The process is then iterated on this smaller automaton. Trahtman's method for finding the sequence of flips leading to a stable pair has a worst-case quadratic time complexity, which makes his algorithm cubic.

In this paper, we design a worst-case linear time algorithm for finding a sequence of flips until the automaton has a stable pair. This makes the algorithm for computing a synchronized coloring quadratic in time and linear in space. The sequence of flips is obtained by fixing a color, say red, and by considering the red cycles formed with red edges, taking into account the positions of the roots of red trees attached to each cycle. The price to pay for decreasing the time complexity is some more complication in the choice of the flips. We also extend the Road Coloring Theorem to periodic graphs by showing that Trahtman's algorithms provide a minimal-rank coloring. Another proof of this result using semigroup tools, obtained independently, is given in [7]. For related results, see also [30,17].

The complexity of synchronization problems on automata has been already studied (see [20] for a survey). It is well-known that there is an $O(n^2)$ algorithm to test whether an n-state automaton on a fixed-size alphabet is synchronized. The complexity of computing a specific synchronizing word is $O(n^3)$ (see [14]). However, the complexity of finding a synchronizing word of a given length is NP-complete [14] (see also [24,27]). The complexity of problems on automata has also been studied for random automata (see [8]). Several results prove that, under appropriate hypotheses, a random irreducible automaton is synchronized [15,28,22]. The average time complexity of these problems does not seem to be known. In particular, we do not know the average time complexity of the Road Coloring Problem.

The article is organized as follows. In Section 2, we give some definitions to formulate the problem in terms of finite automata instead of graphs. In Section 3 we describe Trahtman's algorithm and our variant is detailed in Section 4. We give both an informal description of the algorithm with pictures illustrating the constructions, and a pseudocode. The time and space complexity of the algorithm are analyzed in Section 5. The periodic case is treated in Section 6. A preliminary version of this paper was posted in [3].

2. The Road Coloring Theorem

In order to formulate the *Road Coloring Problem* we introduce the notation concerning automata.

Let A be a finite alphabet and let Q be a finite set. We denote by A^* the set of words over A.

A (finite) *automaton* $\mathcal{A} = (Q, E)$ over the alphabet A with Q as set of states is a given by a set E of edges which are triples (p, a, q) where p, q are states and a is a symbol from A called the label of the edge. Note that no initial or final states are specified. Let F be the multiset formed of the pairs (p, q) obtained from the set E by the map $(p, a, q) \mapsto (p, q)$. The multigraph having Q as set of vertices and F as set of edges is called the *underlying graph* of $\mathcal{A}$.

A *path* in the automaton is the sequence of consecutive edges. The label of the path $(p_i, a_i, p_{i+1})_{0 \le i \le n}$ is the word $a_0 \ldots a_n$. The state p_0 is its origin and p_{n+1} is its end. The *length* of the path is $n + 1$. The path is a *cycle* if $p_0 = p_{n+1}$.

An automaton is *deterministic* if, for each state p and each letter a, there is at most one edge starting at p and labeled with a. It is *complete deterministic* if, for each state p and each letter a, there is exactly one edge starting at p and labeled with a. This implies that for each state p and each word w there is exactly one path starting at p and labeled with w. The end of this unique path is denoted by $p \cdot w$.

An automaton is *irreducible* if its underlying graph is strongly connected. The *period* of an automaton is the gcd of length of its cycles. An automaton is *aperiodic* if it is irreducible and of period 1.[1]

A *synchronizing word* of a complete deterministic automaton $\mathcal{A} = (Q, E)$ is a word $w \in A^*$ such that for every pair of states $p, q \in Q$, one has $p \cdot w = q \cdot w$. A synchronizing word is also called a *reset sequence* [14], or a *magic sequence* [5,6], or also a *homing word* [26]. An automaton which has a synchronizing word is called *synchronized* (see an example on the right part of Fig. 1).

Two automata which have isomorphic underlying graphs are called *equivalent*. Hence two equivalent automata differ only by the labeling of their edges. In the sequel, we shall consider only complete deterministic automata.

Proposition 1. *A synchronized complete deterministic automaton is aperiodic.*

Proof. We assume that the automaton has at least one edge. Let (p, a, q) be an edge of the automaton. Let w be a synchronizing word focusing to a state r. Since the graph is strongly connected, there is a word v such that from $r \cdot v = p$. Thus $p \cdot awvp = p \cdot wvp$. The lengths of the cycles from p to p labeled awv and wv differ by 1. This implies that the period of automaton is 1. □

---

1 Note that this notion, which is usual for graphs, is not the notion of aperiodic automata used elsewhere and which refers to the period of words labeling the cycles (see *e.g.* [13]).