On Recognizing Cayley Graphs*

Lali Barrière[1], Pierre Fraigniaud[2], Cyril Gavoille[3], Bernard Mans[4], and John M. Robson[3]

[1] Departament de Matemàtica Aplicada i Telemàtica, *Univ. Politècnica de Catalunya*, Campus Nord, Edifici C3, c/ Jordi Girona 1-2, 08034 Barcelona, Spain.
[2] Laboratoire de Recherche en Informatique, Bât. 490, *Univ. Paris-Sud*, 91405 Orsay cedex, France.
[3] Laboratoire Bordelais de Recherche en Informatique, *Univ. Bordeaux I*, 33405 Talence cedex, France.
[4] Department of Computing, *Macquarie Univ.*, Sydney, NSW 2109, Australia.

Abstract. Given a class $\mathcal{C}$ of Cayley graphs, and given an edge-colored graph G of n vertices and m edges, we are interested in the problem of checking whether there exists an isomorphism ϕ preserving the colors such that G is isomorphic by ϕ to a graph in $\mathcal{C}$ colored by the elements of its generating set. In this paper, we give an $O(m \log n)$-time algorithm to check whether G is color-isomorphic to a Cayley graph, improving a previous $O(n^{4.752} \log n)$ algorithm. In the case where $\mathcal{C}$ is the class of the Cayley graphs defined on Abelian groups, we give an optimal $O(m)$-time algorithm. This algorithm can be extended to check color-isomorphism with Cayley graphs on Abelian groups of given rank. Finally, we propose an optimal $O(m)$-time algorithm that tests color-isomorphism between two Cayley graphs on $\mathbb{Z}_n$, i.e., between two circulant graphs. This latter algorithm is extended to an optimal $O(n)$-time algorithm that tests color-isomorphism between two Abelian Cayley graphs of bounded degree.

1 Introduction

Checking whether two input graphs with n vertices and m edges are isomorphic is known as the *graph-isomorphism* problem and is a famous combinatoric problem. For general graphs the graph-isomorphism problem is known to be in NP, not known to be in P, and probably is not NP-complete (see Section 6 of [3]). A relaxed problem is to recognize classes of graphs, that is to check whether a given graph G is isomorphic to an element of a specified class of graphs (e.g., interval graphs, chordal graphs, planar graphs, etc.). Depending on the class of graphs, this problem can be easy (e.g., checking planarity). However, for many classes, checking whether a graph belongs to that class remains difficult. This is particularly true for Cayley graphs [12].

* This work is supported by an Australian-French cooperation granted by CNRS and ARC. Part of this work has been done while the three first authors were visiting the Department of Computing of Macquarie University, Sydney. Additional support by Spanish Research Council (CICYT) under project TIC97-0963, by the CNRS, and by the Aquitaine Region project #98024002.

A graph $G = (V, E)$ is *Cayley* if there exists a group Γ and a generating set S of Γ such that V is the set of the elements of Γ, and there is an arc from x to y in G, that is $(x, y) \in E$, if and only if there exists $s \in S$ such that $y = s * x$, or, in other words, if and only if $y * x^{-1} \in S$. (In the following, we will use the additive notation for the operation $*$ of Γ). If G is Cayley, then G is denoted by the pair (Γ, S). If S is involutive, that is if $(s \in S \Rightarrow s^{-1} \in S)$ holds, then G is a symmetrically-oriented directed graph which can be seen as an undirected graph. The regularity and the underlying algebraic structure of Cayley graphs make them good candidates for interconnecting nodes of a network [11]. For instance, Cycles, Tori, Hypercubes, Cube-Connected Cycles (CCC), Butterflies with wraparound, Star-graphs, Bubble Sort graphs, Pancake graphs, Chordal rings (i.e., circulant graphs) are all Cayley graphs. Some of them are defined on Abelian groups (e.g., hypercubes and tori), whereas some others are defined on non-Abelian groups (e.g., Star-graphs, and Pancake graphs).

The complexity of graph-isomorphism for Cayley graphs is a challenging problem [12, 14]. Currently, the problem is still open even if we restrict our attention to circulant graphs [2, 7], i.e., Cayley graphs defined on $\mathbb{Z}_n$. The problem is known to be polynomial only for specific classes of circulant graphs, e.g., those with a prime number of vertices [6, 15]. In this paper, we study a variant of the graph-isomorphism problem. Given two (properly or not) arc-colored digraphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, these two digraphs are said *color-isomorphic* if there exists an isomorphism from G_1 to G_2 that preserves the colors. More formally, let C_1 be the set of colors of G_1, and C_2 be the set of colors of G_2. G_1 and G_2 are color-isomorphic if there exist a one-to-one mapping $\phi : V_1 \to V_2$ and a one-to-one mapping $\psi : C_1 \to C_2$ such that for every $c \in C_1$, and for every $(x, y) \in V_1 \times V_1$:

$$(x, y) \in E_1 \text{ and is of color } c \iff (\phi(x), \phi(y)) \in E_2 \text{ and is of color } \psi(c) \ .$$

Graph-isomorphism is the particular case of the color-isomorphism problem obtained by choosing $C_1 = C_2 = \{c\}$. Conversely, one can construct a polynomial reduction from color-isomorphism to graph-isomorphism[1]. Therefore, the two problems are equivalent. That is why we will focus on *specific* colorings of graphs to solve the color-isomorphism problem. In particular, the generating set of a Cayley graph (Γ, S) induces a coloring of the edges, simply by coloring the edge $(x, s + x)$ by the element $s \in S$. Such a coloring is called the *natural* edge-coloring of a Cayley graph. As we show in this paper, the color-isomorphism problem between naturally colored Cayley graphs is apparently simpler than graph-isomorphism, or color-isomorphism, between arbitrary colored Cayley graphs.

The notion of color-isomorphism applied to naturally colored Cayley graphs was introduced to characterize edge-labeled regular graphs that have so-called *minimal sense of direction* [8]. In [9] it is shown that such regular graphs are

[1] In fact, we will show in the full version of the paper that there is a polynomial reduction from color-isomorphism of Cayley graphs to Cayley isomorphism.

exactly those that are color-isomorphic to a Cayley graph whose edges are colored by the elements of its generating set. Boldi and Vigna proposed in [5] an $O(n^{4.752} \log n)$-time algorithm to check whether a colored graph is color-isomorphic to a naturally colored Cayley graph. A parallel version of this algorithm runs in $O(\log n)$ time with n^6 processors.

Note that there are major differences between, on one hand, the problem of checking whether a given graph is isomorphic (or color-isomorphic) to a Cayley graph of a specified class, and, on the other hand, the problem of checking whether a given group is isomorphic to a group of a specified class [13]. For instance, an $O(n)$-time algorithm exists for Abelian group isomorphism [16], but the addition table of the group is given as input of the problem. In the color-isomorphism problem, the algebraic structure of the graph is hidden, and it may not be possible to perform elementary group operations in constant time. In this paper we show how to extract the structure of the underlying group of colored Cayley graphs. This is done by a pure graph and algorithmic approach.

Results and structure of the paper

1) In Section 3, we present an $O(m \min\{k, \log n\}) = O(m \log n)$-time algorithm that checks whether a colored graph of degree k given as input is color-isomorphic to a naturally colored Cayley graph, improving a previous $O(n^{4.752} \log n)$ algorithm.

2) In Section 4, we give an $O(m)$-time algorithm that checks whether a colored graph given as input is color-isomorphic to a naturally colored Cayley graph defined on an Abelian group. This algorithm trivially extends to an algorithm that, given any integer r and any graph G, checks whether G is color-isomorphic to a naturally colored Cayley graph on an Abelian group of rank r. As a consequence, we can check in $O(m)$-time whether a colored graph given as input is color-isomorphic to a naturally colored circulant graph ($r = 1$). The decrease of the complexity from $O(m \log n)$ for arbitrary Cayley graphs to $O(m)$ for Abelian Cayley graphs is obtained by a sophisticated technique used to perform arithmetic computations in $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_r}$, $\prod_{i=1}^{r} n_i = n$, with the standard $O(\log n)$-word RAM model [10].

3) Finally, in Section 5, we give an $O(m)$-time algorithm that checks color-isomorphism between two naturally colored circulant graphs given as input. This algorithm can be generalized to produce an $O(n)$-time algorithm that checks color-isomorphism between two bounded degree naturally colored Abelian Cayley graphs.

Note that our $O(m)$-time algorithms are optimal since, in the worst-case, we have to check the m edges of the input graph. As far as the sense of direction theory is concerned, our results have the following consequence:

Corollary 1. *There is an $O(m \min\{k, \log n\})$-time algorithm which, given any regular graph and any edge-coloring of this graph, returns whether this coloring gives G a minimal sense of direction.*

The next section gives preliminary results that are helpful for solving our problems.

2 Notation and Preliminary Results

The model. In each of our problems, we assume that the input graph is a digraph (that is a strongly connected directed graph). Again, a graph is a particular case of this setting by viewing the graph as a symmetrically-oriented directed graph. The input arc-colored digraph of order n is described by its adjacency list $\mathcal{L}$, including the colors. The vertices are labeled from 1 to n, and, for every vertex x, $\mathcal{L}[x]$ is the list $((y_1, c_1), \ldots, (y_k, c_k))$ of the k out-neighbors of x together with the colors of the arcs linking x to these out-neighbors. Each color c_i is an integer taken in the set of colors $C = \{1, \ldots, k\}$. Actually, for k-regular digraphs (that is where each vertex is of in- and out-degree k), the list can be replaced in $O(nk)$ time by a $n \times k$ *adjacency table* $\mathcal{T}$ so that, for every vertex x and for every $i \in \{1, \ldots, k\}$, $\mathcal{T}[x, c_i]$ returns in constant time the i-th neighbor y_i of x such that the arc (x, y_i) is colored c_i. Therefore, for a k-regular digraph of n vertices, the size of the input is nk integers taken from the set $\{1, \ldots, n\}$.

Note that one can check in $O(nk)$ time whether a digraph is k-regular, and whether a digraph is properly colored by k colors (that is every vertex has its k incoming arcs colored with k different colors, and its k outgoing arcs colored with the same k different colors). Therefore, unless specified otherwise, we will always assume in the following that the input digraph is k-regular, and properly colored with k colors.

Notation. Let x_0 be an arbitrary vertex of a properly colored digraph $G = (V, E)$. Since we are checking whether G is isomorphic to a subclass of Cayley digraphs, one can assume that G is vertex-transitive. Indeed, if this assumption happens to be false, it will create a contradiction further at some stage of the algorithm, and therefore it will be detected. Thus, we can assume, w.l.o.g., that x_0 is the zero of the underlying group. As stated previously, the elementary group operations are not directly available to be performed in constant time given the digraph G only. Indeed, the (supposed) algebraic structure of the digraph is hidden in its adjacency table. So, let us redefine standard group operations applied to digraphs.

The concatenation of two words w and w' whose letters are taken in the set of colors C, that of two words w and w' of C^*, is denoted by $w \otimes w'$. Let $\mathcal{S}$ be any directed tree spanning G, and rooted at x_0. We can label each vertex x of G by a sequence of colors according to the color of the arcs encountered in the path from x_0 to x in $\mathcal{S}$. More precisely x_0 is labeled $\ell(x_0) = \epsilon$ (the empty sequence), and every vertex $x \neq x_0$ is labeled $\ell(x) = \ell(y) \otimes c$, i.e., the concatenation of the label of the father y of x in $\mathcal{S}$, and of the color c of the arc (y, x). Once this labeling is fixed, one can define the *orbit* of a vertex $x \neq x_0$ as the ordered sequence of vertices $y_0, y_1, y_2, \ldots$ where $y_0 = x_0$, and, for $i > 0$, y_i is the vertex reached from y_{i-1} by following the sequence of colored arcs given by the sequence of colors of $\ell(x)$. In particular $y_1 = x$. G being properly colored, one can then define the *order* of a vertex $x \neq x_0$ as the smallest integer $t > 0$ for which $y_t = x_0$. The order of x is denoted by $\mathbf{o}(x)$, and $\mathbf{o}(x_0)$ is set to 1.

By extension, the order of a color c, denoted by $\mathbf{o}(c)$, is defined as the order of the out-neighbor x_c of x_0 such that the arc (x_0, x_c) is colored c. The orbit of c is the orbit of x_c. Note that the orbit of a color is an elementary cycle in G, whereas the orbit of an arbitrary vertex x may not be a cycle, in particular if x is not an out-neighbor of x_0. Finally, we define the *addition* of vertices. Given two vertices x and y, $x + y$ is defined as the vertex reached from x by following the path $\ell(y)$. The orbit of a vertex x *passing through* $y \in V$ is simply the sequence $y, y + x, y + 2x, \ldots$

Correspondence tables. Computing the order of a color c can be done in $O(\mathbf{o}(c))$ time by simply following from x_0 the path composed of arcs of color c. Adding a color c to any vertex y takes $O(1)$ time using the adjacency table $\mathcal{T}$, because the vertex $y + c$ is given by $\mathcal{T}[y, c]$. However, computing the order of an arbitrary vertex x looks harder since following the path of color $\ell(x)$ from a vertex y requires $|\ell(x)|$ accesses to the adjacency table, where $|\ell(x)|$ denotes the length of sequence $\ell(x)$. Nevertheless, this complexity can be reduced if we restrict our study to the case $\ell(x) = c \otimes \cdots \otimes c$, with $|\ell(x)| = \alpha$. One can compute the vertex $y + \alpha c$ in $O(\alpha)$ time using $\mathcal{T}$, but one can do better as shown in the following lemma:

Lemma 1. *There is an $O(nk)$-time and $O(nk)$-space algorithm that, given a properly colored k-regular digraph G of order n described by its adjacency table $\mathcal{T}$, either returns* FALSE*, and then G is not a Cayley digraph, or returns a data structure such that, if G is color-isomorphic to a naturally colored Cayley digraph (Γ, S), then it enables computation of vertex $x + \alpha s$ in constant time, for any $x \in \Gamma$, any integer $\alpha \geq 1$ and any $s \in S$.*

Proof. Let us first assume that G is a naturally colored Cayley digraph (Γ, S) of order n given by its adjacency table $\mathcal{T}$. First, let us describe the structure, and show how to compute $z = y + \alpha c$ in constant time using that structure, that is a vertex reached from y by following the path $\ell(\alpha c) = c \otimes \cdots \otimes c$ composed of α times the arc colored by c. Then we will show how this structure can be constructed in $O(nk)$ time. For every $c \in C$, that is for every color of the input digraph, G can be viewed as $n_c = n/\mathbf{o}(c)$ c-colored orbits of length $\mathbf{o}(c)$ with arcs of the other colors between the orbits. Let us concentrate on the n_c orbits of c. Arbitrarily order these orbits from 1 to n_c, and, inside each orbit, order the vertices of this orbit from 0 to $\mathbf{o}(c) - 1$ by starting at any vertex, and by respecting the natural order induced by the orbit. We get an $(n \times k)$-table T, and k tables T_c, one for each color $c \in C$, of respective size $n_c \times \mathbf{o}(c)$. $T_c[i, j]$ is the vertex x at the j-th position in the i-th orbit of c. Conversely, $T[x, c]$ returns the pair (i, j). Computing $z = y + \alpha c$ can be done in constant time using the table T and the tables T_c, $c \in C$, as follows:

- $T[y, c]$ specifies that y is the j-th element of the i-th orbit of c;
- $z = T_c[i, (j + \alpha) \bmod \mathbf{o}(c)]$.

The construction of T and $T_c, c \in C$, can be done in $O(nk)$ time as follows. There is one phase for each color $c \in C$. Once c is fixed, we visit all the vertices

and construct the n_c orbits of c (the visit of the first orbit allows to set $\mathbf{o}(c)$). This is done in n_c steps, one step for each orbit. At each step, say at step i, an arbitrary not yet visited vertex x is chosen. $T[x,c]$ is set to $(i,0)$, and $T_c[i,0]$ is set to x. Visiting the $\mathbf{o}(c)-1$ remaining vertices $x_1,\ldots,x_{\mathbf{o}(c)-1}$ of the current orbit allows to set $T[x_j,c]$ to (i,j), and $T_c[i,j]$ to x_j, for all j, $1 \leq j < \mathbf{o}(c)$. The total cost of this construction is $O(nk)$ because there are $k = |C|$ phases, and, at each phase, every vertex is visited exactly once because G is a Cayley digraph.

Given an arbitrary properly colored k-regular digraph G, one can apply the previous construction. This construction may fail if G does not have enough symmetry, and, conversely, it may succeed even if G is not a Cayley digraph. Nevertheless, for any digraph such that, for every color c, the set of vertices can be decomposed in n_c c-colored orbits of the same length, the construction will succeed. Since G is properly colored, the only thing that must be checked is whether all orbits of the same color have the same length. The algorithm returns FALSE if it fails. This checking can be done on the fly, during the construction of the tables, with no additional cost. □

Definition The tables resulting of the proof of Lemma 1 are called the *correspondence tables* of the colored digraph G.

3 Recognizing Cayley Colored Digraphs

Recall that $C = \{1,\ldots,k\}$ is the set of colors. For any vertex x, we define the equivalence relation $\equiv_x$ on words in C^* by $w_1 \equiv_x w_2$ if and only if $x+w_1 = x+w_2$. Then we define the equivalence relation $\sim$ on vertices by $x_1 \sim x_2$ if and only if $\equiv_{x_1}$ and $\equiv_{x_2}$ are identical.

Lemma 2. *A k-regular connected properly colored digraph $G = (V,E)$ is a Cayley digraph if and only if $v_1 \sim v_2$ holds for every pair (v_1,v_2) of vertices of V.*

Proof. The "only if" part is trivial: if the graph is Cayley, then for every vertex v, the relation $\equiv_v$ is exactly the relation under which words are equivalent if and only if their values in the group are equal. We write $\equiv$ for the relation $\equiv_v$ of every vertex $v \in V$. Choose a vertex $x_0 \in V$, choose a spanning tree rooted at x_0, and define $\ell(x)$ for every vertex $x \in V$ as in Section 2. Recall the definition of $x_1 + x_2$ as $x_1 + \ell(x_2)$ in Section 2. It is easy to see that x_0 is an identity for this $+$ and that if w is the label of a path from x_1 to x_0, then $x_0 + w$ is an inverse of x_1. We have that $\ell(x_1) \otimes \ell(x_2) \equiv \ell(x_1 + x_2)$ since both are the colors of paths from x_0 to $x_1 + x_2$. Now the operation $+$ is associative since $\ell(x_1 + (x_2 + x_3)) \equiv \ell(x_1) \otimes (\ell(x_2) \otimes \ell(x_3))$ and $\ell((x_1 + x_2) + x_3) \equiv (\ell(x_1) \otimes \ell(x_2)) \otimes \ell(x_3)$. □

Lemma 3. *If $x_1 \sim x_2$, then $x_1 + c \sim x_2 + c$.*

Proof. Suppose that $x_1 \sim x_2$. If $w_1 \equiv_{x_1+c} w_2$, then $(x_1 + c) + w_1 = (x_1 + c) + w_2$ so $x_1 + (c \otimes w_1) = x_1 + (c \otimes w_2)$. In other words, $c \otimes w_1 \equiv_{x_1} c \otimes w_2$ and so $c \otimes w_1 \equiv_{x_2} c \otimes w_2$ giving $x_2 + (c \otimes w_1) = x_2 + (c \otimes w_2)$ and finally $(x_2 + c) + w_1 = (x_2 + c) + w_2$ so that $w_1 \equiv_{x_2+c} w_2$. □

The following $O(m)$-time algorithm tests for two vertices x and y whether $x \sim y$:

Testing equivalence algorithm

1. Choose a spanning tree rooted at x and label the vertices $x = x_0, x_1, \ldots, x_{n-1}$ according to a pre-order traversal. Traverse the similar tree rooted at y (i.e., using the same colors as in the tree rooted at x), checking that no vertex is encountered twice and labeling the vertices $y = y_0, y_1, \ldots, y_{n-1}$ again in pre-order.
2. For every $i < n$ and every color $c \in C$, consider $x_i + c = x_j$ and $y_i + c = y_{j'}$; if in any case $j \neq j'$ then return FALSE, else return TRUE.

Proof. Let w_i be the word in C^* labeling the path in the spanning tree from x to x_i. If a vertex is encountered twice in traversing the tree rooted at y, say in the positions corresponding to x_i and x_j then we have that $w_i \equiv_y w_j$ but $w_i \not\equiv_x w_j$; if we ever found $x_i + c = x_j$, $y_i + c = y_{j'}$ with $j \neq j'$, then $w_i \otimes c \equiv_x w_j$ but $w_i \otimes c \not\equiv_y w_j$. On the other hand, if we always have $j = j'$, we have a way of reducing any word to a unique equivalent w_i which is valid for both $\equiv_x$ and $\equiv_y$. □

Given Lemma 3, we will know that all vertices are equivalent to x_0 if we find a set $X = \{c_1, \ldots, c_t\}$ of colors such that $x_0 \sim x_0 + c_i$ $(1 \leq i \leq t)$ and every vertex is reachable from x_0 by a path labeled by colors from X. If the graph is Cayley, each new color added to X takes us from a subgroup to a larger subgroup and so at least doubles the number of reachable vertices. Hence at most $\min\{k, \log n\}$ colors can be added before either exhausting V or finding that the graph is not Cayley.

Theorem 1. *There is an $O(m \min\{k, \log n\})$-time algorithm which, given a properly colored k-regular digraph G, checks whether G is color-isomorphic to a naturally colored Cayley digraph.*

Proof. Starting with a set S of vertices including only x_0, and a set of colors X initially empty, we consider each color $c \in C$ in turn. If $x_0 + c$ is already in S we do nothing. Otherwise we test that $x_0 + c \sim x_0$, rejecting if not, add c to X and then add to S all vertices reachable from existing elements of S by a path starting with color c and then using any colors from X. If this exhausts V accept and otherwise if the set of colors is exhausted reject.

Given that we stop the recursive process of adding vertices to S when we encounter a vertex already in S, any edge is only considered once and so the time is dominated by the at most $\min\{k, \log n\}$ tests for $x_0 + c \sim x_0$ each of which takes time $O(m)$ giving the total time bound of $O(m \min\{k, \log n\})$.

Note that we do not need to test that $|S|$ has at least doubled after each incremental operation because the subgraph consisting of the vertices in S and the edges between them colored by colors in X is always a Cayley graph. □

We were not able to prove or disprove the optimality of our algorithm. It is obviously optimal for bounded degree graphs, but there is a $O(\log n)$ gap between its worst-case complexity and the trivial lower bound $\Omega(m)$ for arbitrary graphs. In the next section, we give an optimal algorithm for the class of Abelian Cayley graphs.

4 Recognizing Colored Cayley Digraphs on Abelian Groups

Recall that the class of Abelian Cayley graphs contains networks as popular as the hypercubes, the tori, and the circulant graphs. Let $n = \prod_{i=1}^{t} p_i^{m_i}$ be the prime-decomposition of n. An Abelian group Γ of order n is determined by a specific decomposition of the m_i's in $m_i = \sum_{j=1}^{r_i} \alpha_{i,j}$, $\alpha_{i,1} \geq \ldots \geq \alpha_{i,r_i} \geq 1$. We then get (see for instance [4]):

$$\Gamma \simeq (\mathbb{Z}_{p_1^{\alpha_{1,1}}} \times \cdots \times \mathbb{Z}_{p_1^{\alpha_{1,r_1}}}) \times (\mathbb{Z}_{p_2^{\alpha_{2,1}}} \times \cdots \times \mathbb{Z}_{p_2^{\alpha_{2,r_2}}}) \times \cdots \times (\mathbb{Z}_{p_t^{\alpha_{t,1}}} \times \cdots \times \mathbb{Z}_{p_t^{\alpha_{t,r_t}}}) \quad (1)$$

By ordering the $\alpha_{i,j}$'s so that $\alpha_{i,j} \geq \alpha_{i,j+1}$, and the p_i's so that $p_i > p_{i+1}$, let

$$n_j = \prod_{i=1}^{t} p_i^{\alpha_{i,j}},\ j = 1, \ldots, r,$$

where $r = \max_i r_i$, and $\alpha_{i,j} = 0$ for $j > r_i$. The rank of Γ is then equal to r. Note that, for any Cayley digraph $G = (\Gamma, S)$ to be connected, we necessarily have $r \leq k = |S|$. From the definition of the n_j's, one can express Γ in the form $\mathbb{Z}_{n_1} \times \cdots \times \mathbb{Z}_{n_r}$, $n_i \geq n_{i+1}$. (Note that n_{i+1} divides n_i.) This latter form is called the *standard form* of Γ.

Our recognition algorithm performs in three phases. The first finds a basis of the unknown $\mathbb{Z}$-module Γ corresponding to the input digraph G. Once this basis has been determined, that is once the group Γ has been identified, we try, during the second phase, to label the vertices, and to compute a candidate for S such that $G = (\Gamma, S)$. After that, a one-to-one mapping between the vertex-sets of the two graphs has been set. Therefore, the third phase checks whether or not the colors of the edges of G and (Γ, S) correspond by a one-to-one mapping. We actually do not label directly the vertices of G by r-tuples because it is not computationally cheap to find elements of order $n_1, \ldots, n_r$. Rather, we still first make use of Γ in its original form depicted by Eq. (1). This will yield labels of $\sum_{i=1}^{t} r_i$ coordinates. We will refine this construction later, to finally get labels on r coordinates.

Notation. We will denote by $\langle x_1, \ldots, x_t \rangle$ the *orbit* of $x_1, \ldots, x_t$, that is the set of the linear combinations of $x_1, \ldots, x_t$, no matter if the x_i's are vertices of a digraph, or elements of a group.

In the statement of the following lemma, x_0 refers to the root of the tree $\mathcal{S}$ defined in Section 2.

Lemma 4. *There is an $O(k\sqrt{n}+n)$-time algorithm which, given a properly colored k-regular digraph G and its correspondence tables, returns*

1. *a t-tuple $(r_1,\ldots,r_t)$;*
2. *t r_i-tuples $(\alpha_{i,1},\ldots,\alpha_{i,r_i})$, $i\in\{1,\ldots,t\}$; and*
3. *$\sum_{i=1}^{t} r_i$ vertices $g_{i,j}$, $i\in\{1,\ldots,t\}$, $j\in\{1,\ldots,r_i\}$;*

such that, if G is color-isomorphic to a naturally colored Abelian Cayley digraph on Γ, then

- $\mathbf{o}(g_{i,j}) = p_i^{\alpha_{i,j}}$ *for every i and j;*
- $\langle g_{i,j}\rangle \cap \langle g_{i,1},\ldots,g_{i,j-1}\rangle = \{x_0\}$ *for every i, and every $j>1$;*
- $\langle g_{i,1},\ldots,g_{i,r_i}\rangle \cap \langle g_{i',1},\ldots,g_{i',r_{i'}}\rangle = \{x_0\}$ *for every $i\neq i'$; and*
- $\Gamma \simeq (\mathbb{Z}_{p_1^{\alpha_{1,1}}}\times\cdots\times\mathbb{Z}_{p_1^{\alpha_{1,r_1}}})\times\cdots\times(\mathbb{Z}_{p_t^{\alpha_{t,1}}}\times\cdots\times\mathbb{Z}_{p_t^{\alpha_{t,r_t}}})$.

The proof of Lemma 4 will be given in the full version of the paper.

Theorem 2. *There is an $O(m)$-time and $O(m)$-space algorithm which, given a colored digraph G of m arcs, checks whether G is color-isomorphic to a naturally colored Cayley digraph on an Abelian group. In the affirmative, the algorithm returns the corresponding Cayley digraph (Γ,S), Γ given in the standard form $\Gamma=\mathbb{Z}_{n_1}\times\cdots\times\mathbb{Z}_{n_r}$, and the two one-to-one mappings $\phi: V\to\Gamma$ and $\psi: C\to S$.*

Proof. As said before, we first check in $O(m)$ time that G is properly colored and regular. If it succeeds we apply Lemma 1 and Lemma 4, otherwise return FALSE and stop. To label the vertices, we will not directly use the $g_{i,j}$'s derived in Lemma 4, but we will group some of them together to get a system of r coordinates rather than $\sum_{i=1}^{t} r_i$ coordinates. Let

$$h_j = \sum_{i=1}^{t} g_{i,j},\quad j=1,\ldots,r$$

where $g_{i,j}=0$ if $j>r_i$. We get $\mathbf{o}(h_j)=n_j$ for every j, and we label the vertices in $\Gamma=\mathbb{Z}_{n_1}\times\cdots\times\mathbb{Z}_{n_r}$ as follows.

Vertex x_0 is labeled by the r-tuple $(0,0,\ldots,0)$. Vertices s_i's are simply represented by the neighbors of x_0. The β_1-th vertex of the orbit of h_1 is labeled the r-tuple $(\beta_1,0,\ldots,0)$. The β_2-th vertex of the orbit of h_2 passing through $(\beta_1,0,\ldots,0)$ (starting from $(\beta_1,0,\ldots,0)$) is labeled $(\beta_1,\beta_2,0,\ldots,0)$. And so on. Thanks to Lemma 1, each jump along the orbit of h_1 (starting at x_0) can be computed in $O(\min\{k,t\})$ time since h_1 is the sum of multiples of at most $\min\{k,t\}$ colors. Similarly, each jump along the orbit of h_{i+1}, starting at $(\beta_1,\beta_2,\ldots,\beta_i,0,\ldots,0)$ takes a time $O(\min\{k,it\})$ because h_{i+1} is the sum of multiples of at most $O(\min\{k,it\})$ colors. Therefore, the whole labeling can be performed in $O(n\min\{k,rt\})$ time, that is in at most $O(m)$ time. Meanwhile, the sequence $(n_1,\ldots,n_r)$ has been computed in at most $O(m)$ time.

Once the labeling has been performed, it remains to check that, for every $i=1,\ldots,k$, all arcs

$$(x,y) = ((x_1,\ldots,x_r),(y_1,\ldots,y_r))$$

colored c_i satisfy

$$\begin{cases} y_1 - x_1 = \psi_1(c_i) \pmod{n_1}; \\ y_2 - x_2 = \psi_2(c_i) \pmod{n_2}; \\ \qquad\vdots \\ y_r - x_r = \psi_r(c_i) \pmod{n_r}; \end{cases} \tag{2}$$

where $\psi(c) = (\psi_1(c), \ldots, \psi_r(c))$ is a r-tuple of $\mathbb{Z}^r$ depending only on $c \in \{c_1, \ldots, c_k\}$. If these equalities are satisfied by every color c_i, then G is color-isomorphic to (Γ, S) where $S = \{\psi(c_1), \ldots, \psi(c_k)\}$. Otherwise, G is not color-isomorphic to a naturally colored Cayley digraph on an Abelian group (and we return FALSE and stop). Indeed, the map $\phi : \Gamma \to \Gamma$ such that $\phi(a_1h_1 + \ldots + a_rh_r) = (a_1, \ldots, a_r)$ is a group-automorphism, and therefore, if $G = (\Gamma, \{s_1, \ldots, s_k\})$, then $\psi(c_i)$ is actually the labeling of s_i, that is $\psi(c_i) = \phi(s_i) = \phi(v^{(i)})$ where $v^{(1)}, \ldots, v^{(k)}$ are the k neighbors of x_0 such that, for every $i \in \{1, \ldots, k\}$, $v^{(i)} = s_i + x_0$. A naive component-wise approach to check Eq. (2) yields an $O(nkr)$-time algorithm with a $O(nk)$-space. As detailed in the full version of the paper Eq. (2) can be checked in $O(1)$ time for each arc by using pre-computed $O(nk)$-space tables requiring $O(nk)$ time to be set. This can be done with the standard $O(\log n)$-word RAM model [10]. □

Recognizing colored Abelian Cayley digraphs of a given rank r can be done by a simple variant of the algorithm of Theorem 2. Indeed, once the $g_{i,j}$'s have been computed in $O(m)$ time (see Lemma 4), the rank of the group Γ such that G is possibly color-isomorphic to a Cayley digraph on Γ is determined. Therefore, one can stop the algorithm if this rank is different from r. The algorithm will perform in $O(m)$ time as shown in Theorem 2. As a consequence:

Corollary 2. *For any integer $r > 0$, there is an $O(m)$-time and $O(m)$-space algorithm which, given a colored digraph G, checks whether G is color-isomorphic to a naturally colored Cayley digraph on an Abelian group of rank r. In the affirmative, the algorithm returns*

- *the corresponding Cayley digraph (Γ, S) where Γ given in its standard form;*
- *the labeling function $\phi : V \to \Gamma$; and*
- *the coloring function $\psi : C \to S$.*

5 Color-Isomorphism for Circulant and Abelian Cayley Digraphs

Circulant graphs have attracted a lot of attention in the literature from both a theoretical and practical point of view. Indeed, the algebraic structure of circulant graphs is simple but rich, and their regularity make them suitable candidates for connecting nodes of a network. Let us show how one can check whether two given naturally colored circulant graphs are color-isomorphic. The elements of the generating set S of a circulant graph $(\mathbb{Z}_n, S)$ are called *chords*. We say that

two subsets T and S of $\mathbb{Z}_n$ are *proportional* if there is an integer λ such that $\gcd(\lambda, n) = 1$ and $T = \lambda S$. Two circulant digraphs are then said to be *Ádám-isomorphic* if the sets of chords are proportional [1]. One can easily show that Ádám-isomorphism and color-isomorphism between naturally colored circulant digraphs are two equivalent notions, that is:

Lemma 5. *Two naturally colored circulant digraphs are color-isomorphic if and only if they are Ádám-isomorphic.*

Therefore, we get:

Theorem 3. *There is an $O(m)$-time algorithm that determines, for any pair G and H of naturally colored circulant digraphs, whether G is color-isomorphic to H.*

Proof. Let G and H be the two input digraphs, both of order n. Let $S, T \subseteq \mathbb{Z}_n$ be the sets of chords returned by the algorithm of Corollary 2 (with $r = 1$) on G and H respectively. From Lemma 5, G and H are color-isomorphic if and only if there exists an invertible constant λ in $\mathbb{Z}_n$ such that $T = \lambda S$. Let $S = \{s_1, \ldots, s_k\}$, and $T = \{t_1, \ldots, t_k\}$. There are at most k candidates for λ: the solutions of $t_i = \lambda s_1$, $i = 1, \ldots, k$, if exists. Each candidate can be computed in $O(\log n)$ time by Euclid's algorithm. For every candidate, one checks whether $T = \lambda S$ (that can be performed in $O(k)$ time assuming a pre-computed n bitmap encoding the set T). All these operations take $O(n + k(\log n + k)) = O(n + k^2)$ time, which is bounded by the time required by the algorithm of Corollary 2. □

The previous result can be generalized to Abelian Cayley digraphs by noticing that, as a generalization of Lemma 5, two naturally colored Abelian Cayley digraphs (Γ, S) and (Γ, T) are color-isomorphic if and only if there exists an automorphism ϕ of Γ such that $T = \phi(S)$, and for any basis B of the $\mathbb{Z}$-module Γ, $\phi(B)$ is also a basis. Therefore, given two naturally colored Abelian Cayley digraphs G_1 and G_2, one can apply Theorem 2 to compute their respective groups Γ_1 and Γ_2, and their respective generating sets S and T. If $\Gamma_1 \neq \Gamma_2$, then G_1 and G_2 are not color-isomorphic. (This latter checking can be done easily be comparing the two sequences $n_1, \ldots, n_r$ of Γ_1 and Γ_2 since our algorithm returns the group in its standard form.) Otherwise, one can test all the one-to-one functions mapping S to T. Computing one generating subset $S' \subseteq S$ of Γ (which can be done in $O(m)$ time), one can show that there are only $k!/(k - |S'|!) \leq k^r$ mappings to test. Choosing a suitable base B, one can compute $\phi(B)$ and check whether $\phi(B)$ is a basis in $O(m)$ time. In total, it gives rise to an $O(k^r m)$-time algorithm.

Corollary 3. *There is an $O(k^r m)$-time algorithm that determines, for any pair G, H of naturally colored Cayley digraphs on Abelian groups of rank r, whether G is color-isomorphic to H.*

Our algorithm is linear for bounded degree and polynomial for bounded rank. We leave as an open problem the following:

Problem 1. Design a polynomial-time algorithm that checks color-isomorphism between two naturally colored Cayley graphs on Abelian groups of arbitrary rank.

Acknowledgments. The authors are thankful to Nicola Santoro who suggested that they study the color-isomorphism problem. The authors also want to express their deep thanks to Sebastiano Vigna for his comments, and Charles Delorme who gave them many hints on how to attack the $\mathbb{Z}_n$-isomorphism problem.

References

1. A. Ádám. Research problem 2-10. *J. Combin. Theory*, 2:393, 1967.
2. B. Alspach and T. Parsons. Isomorphism of circulant graphs and digraphs. *Discrete Mathematics*, 25:97–108, 1979.
3. L. Babai. Automorphism groups, isomorphism, reconstruction. In R. Graham, M. Grötschel, and L. Lovász, editors, *Handbook of Combinatorics*, vol. 2. Elsevier and MIT Press, 1995.
4. G. Birkhoff and S. MacLane. *Algebra.* Macmillan, 1967.
5. P. Boldi and S. Vigna. Complexity of deciding Sense of Direction. *SIAM Journal on Computing*, 29(3):779–789, 2000.
6. B. Codenotti, I. Gerace and S. Vigna. Hardness Results and Spectral Techniques for Combinatorial Problems on Circulant Graphs. *Linear Algebra Appl.*, 285(1–3):123–142, 1998.
7. B. Elspas and J. Turner. Graphs with circulant adjacency matrices. *J. Comb. Theory*, 9:229–240, 1970.
8. P. Flocchini, B. Mans, and N. Santoro. Sense of direction in distributed computing. In *Proceedings of DISC '98*, LNCS vol. 1499, Springer-Verlag, S. Kutten (Ed.), pages 1–15, 1998.
9. P. Flocchini, A. Roncato, and N. Santoro. Symmetries and sense of direction in labeled graphs. Discrete Applied Mathematics, 87:99-115, 1998.
10. T. Hagerup. Sorting and searching on the word RAM. In *Proceedings of STACS '98*, LNCS vol. 1379, Springer-Verlag, M. Morvan and C. Meinel (Eds.), pp. 366–398, 1998.
11. T. Leighton Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes. Morgan Kaufmann, 1992.
12. C. Li, C. Praeger, and M. Xu. On finite groups with the Cayley isomorphism property. *Journal of Graph Theory*, 27:21–31, 1998.
13. G. Miller. On the $n^{\log n}$ isomorphism technique. In *Proceedings Tenth Annual ACM Symposium on Theory of Computing (STOC)*, pp. 51–58, 1978.
14. J. Morris. Isomorphic Cayley graphs on nonisomorphic groups. *Journal of Graph Theory*, 31:345–362, 1999.
15. M. Muzychuk and G. Tinhoffer. Recognizing circulant graphs of prime order in polynomial time. *The electronic journal of combinatorics*, 3, 1998.
16. N. Vikas. An $O(n)$ algorithm for Abelian p-group isomorphism and an $O(n \log n)$ algorithm for Abelian group isomorphism. *Journal of Computer and System Sciences*, 53:1–9, 1996.