Recognizing Abelian Cayley Colored Digraphs*

Lali Barrière† Pierre Fraigniaud‡ Cyril Gavoille§ Bernard Mans¶

January 16, 2000

Technical Report LRI-1242
Laboratoire de Recherche en Informatique
Univ. Paris-Sud, 91405 Orsay cedex, France

Abstract

Given a class $\mathcal{S}$ of Cayley digraphs, and given a properly colored digraph G of order n and size m, 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 digraph in $\mathcal{S}$ colored by the elements of its generating set. Boldi and Vigna have given an AC^1 parallel algorithm for $\mathcal{S}$ equal to the set of all Cayley digraphs. In this paper, we give an $O(m)$-time algorithm to check whether G is color-isomorphic to a Cayley digraph on $\mathbb{Z}_n$ (i.e., a circulant digraph), and an $O(m \min\{k, \log n\})$-time algorithm to check whether G is color-isomorphic to an abelian Cayley digraph (i.e., on a commutative group) where k is the in- and out-degree of G. More precisely, we give an $O(mr)$-time algorithm to check whether G is color-isomorphic to a Cayley digraph on an abelian group of rank r. If the answer is affirmative, then each of our algorithms returns the group Γ on which G is a Cayley digraph. Our algorithms also partially solve the color-isomorphism problem between abelian Cayley digraphs. We give an $O(m)$-time algorithm that checks the color-isomorphism between two circulant digraphs, and an $O(n)$-time algorithm for constant degree abelian Cayley digraphs.

Keywords: Cayley digraph, circulant digraph, digraph isomorphism.

*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.

†Departament de Matemàtica Aplicada i Telemàtica, *Univ. Politècnica de Catalunya*, Campus Nord, Edifici C3, c/ Jordi Girona 1-2, 08034 Barcelona, Spain. Additional support by Spanish Research Council (CICYT) under project TIC97-0963.

‡Laboratoire de Recherche en Informatique, Bât. 490, *Univ. Paris-Sud*, 91405 Orsay cedex, France. Additional support by the CNRS.

§Laboratoire Bordelais de Recherche en Informatique, *Univ. Bordeaux I*, 33405 Talence cedex, France. Additional support by the Aquitaine Region project #98024002.

¶Department of Computing, *Macquarie Univ.*, Sydney, NSW 2109, Australia.

1 Introduction

Checking whether two input digraphs (by *digraph* we always mean connected directed graphs) 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 consists to recognize classes of digraphs, that is to check whether a given digraph G is isomorphic to an element of a specified class of digraphs (e.g., planar digraphs, outer-planar digraphs, etc.). Depending on the class of digraphs, this problem can be easy (e.g., checking planarity). However, for many classes of digraphs, checking whether a digraph belongs to that class remains difficult. This is particularly true for Cayley digraphs [10, 12].

A digraph $G = (V, E)$ is *Cayley* if and only 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 $y * x^{-1} \in S$. If G is a Cayley digraph, G is denoted by the pair (Γ, S). In the following, we will use the additive notation for the operation "$*$" in Γ. *Circulant* digraphs are Cayley digraphs on $\mathbb{Z}_n = \mathbb{Z}/n\mathbb{Z}$, the additive group of the integers modulo n. In this paper, the elements of the generating set S of a circulant digraph are called *chords*. This terminology is extended to the elements of the generating set of any Cayley digraph.

Checking whether two Cayley digraphs are isomorphic is a challenging problem. Actually, the problem is still open even if we restrict our attention to circulant digraphs. The *Ádám conjecture* [1] stated that two circulant digraphs are isomorphic if and only if the set of chords are *proportional*. We say that two subsets S and T of $\mathbb{Z}_n$ are proportional if there is an integer λ such that $\gcd(\lambda, n) = 1$ and $S = \lambda T$. The conjecture was shown to be wrong [8], although it holds in many cases [2, 7, 13, 14, 15]. Two circulant digraphs are then said to be Ádám-isomorphic if and only if the set of chords of any of the two digraphs can be obtained by multiplying the chords of the other digraph by an invertible constant. Two non-Ádám-isomorphic digraphs can be isomorphic because the isomorphism can mix the chords, sending part of the chords a on chords b, and another part on chords $c \neq b$. This motivated the notion of color-isomorphism[1].

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 and only 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 and only if there exist two one-to-one and onto functions $\Phi : V_1 \mapsto V_2$ and $\psi : C_1 \mapsto C_2$ such that for every $c \in C_1$, and for every $(x, y) \in V_1 \times V_1$:

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

In particular, the generating set S of a Cayley digraph $G = (\Gamma, S)$ induces a coloring of the arcs, simply by labeling the elements of S from 1 to $k = |S|$, and by giving the color s to the arc $(x, s + x)$. Such a coloring will be said the *natural* coloring of the arcs of a Cayley digraph. Actually, the following lemma (see the proof in Appendix A) shows that Ádám-isomorphism and color-isomorphism between naturally-colored circulant digraphs are two equivalent notions.

Lemma 1 *Two naturally-colored circulant digraphs are color-isomorphic if and only if they are Ádám-isomorphic.*

[1]Actually, up to the knowledge of the authors, the notion of color-isomorphism was introduced to characterize digraphs which support a so-called *minimal weak sense of direction* [9]: such digraphs are exactly those that can be made, by a suitable coloring of their arcs, color-isomorphic to a Cayley digraph whose arcs are colored by the elements of its generating set.

Boldi and Vigna [5, 6] have given an AC^1 parallel algorithm with n^6 processors to check whether a given colored digraph G is color-isomorphic to a naturally colored Cayley digraph. This result shows that the problem of checking whether a given arc-colored digraph G is isomorphic to a naturally colored Cayley digraph is significantly simpler than the recognition of Cayley digraphs, in the sense that the former problem can be solved in polynomial time, whereas the complexity of the latter is still unknown. This was the motivation to study the color-isomorphism for other classes of digraphs. In this paper, we concentrate on subclasses of Cayley digraphs. More specifically, given an arc-colored digraph G, we will consider the following problems: Is G color-isomorphic to a naturally colored Cayley digraph on an abelian group, and if yes which is this group? Is G color-isomorphic to a naturally colored Cayley digraph on an abelian group of given rank r, and if yes which is this group? In particular, is G color-isomorphic to a naturally colored circulant digraph? A Cayley digraph on an abelian group will simply be called an abelian Cayley digraph. (In this paper, all groups and graphs are supposed to be of finite order.)

Note that there are major differences between, on one hand, the problem of checking whether a given digraph is isomorphic to a Cayley digraph 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 [11, 16]. In particular, in the former problem, the algebraic structure of the digraph is hidden, and elementary group operations may not be able to be performed in constant time, whereas group operations can be performed in constant time in the latter problem, e.g., if the input group is described by its addition table. Nevertheless, in both cases, there are many concepts that can be described in similar terms, as we will see later. This is allowed by the $\mathbb{Z}$-module structure of any abelian group [4].

Our results.

- We give an $O(m \min\{k, \log n\})$-time algorithm to check whether a k-regular colored digraph of order n and size m is color-isomorphic to a naturally colored abelian Cayley digraph, and, in the affirmative, to explicitly construct such a Cayley digraph (Γ, S).
- We also give an $O(mr)$-time algorithm to check whether a colored digraph is color-isomorphic to a naturally colored abelian Cayley digraph on a group of rank r. (Again, in the affirmative, the algorithm explicitly constructs the Cayley digraph.) In the case $r = 1$, we have derived a specific simple $O(m)$-time algorithm to check whether a colored digraph is color-isomorphic to a naturally colored circulant digraph. This latter algorithm is optimal since we have to check the m arcs of the input digraph.
- Finally, we give an $O(m)$-time algorithm that checks the color-isomorphism between two naturally colored circulant digraphs. This algorithm can be generalized to produce an $O(n)$-time algorithm to check the color-isomorphism of two constant degree naturally colored abelian Cayley digraphs.

Intuitively, the $O(m \min\{k, \log n\})$-time and $O(mr)$-time algorithm should not be too far from the optimal since a somewhat standard way to describe a Cayley digraph on an abelian group $\mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_r}$ of rank r is to give the list of the k r-tuples $(c_{i,1}, \ldots, c_{i,r})$, $i = 1, \ldots, r$, defining the k chords. Our algorithms do nothing else than picking every vertex, and checking the adjacencies of each vertex. Finding a faster algorithm would require to derive techniques that are not based on labeling the vertices, and checking the adjacency. We were not able to discover such an algorithm, neither to disprove its existence, and thus we let this question as an open problem:

Problem 1 *Finding the exact complexity of recognizing naturally colored abelian Cayley digraphs.*

About the graph-isomorphism issue, our $O(n)$-time algorithm to check the color-isomorphism of two constant degree naturally colored abelian Cayley digraphs has its complexity growing exponentially with the degree. We were not able to do better. Therefore, we let as an open problem the following:

Problem 2 *Constructing a polynomial algorithm for checking the color-isomorphism of two naturally colored abelian Cayley digraphs of arbitrary degree.*

Structure of the paper. The paper is organized as follows. The next section presents preliminary results which will be shown to be helpful to solve our problems. Then, Section 3 presents our algorithm checking the color-isomorphism of a given digraph with an abelian Cayley digraph. Section 4 presents our algorithm checking the color-isomorphism of a given digraph with an abelian Cayley digraph of rank r, and in particular with circulant digraph. The same section also contains our results about the color-isomorphism of two given naturally colored abelian Cayley digraphs.

2 Notations and preliminary results

In each of our problems, the input digraph of order n is described by its adjacency list $\mathcal{L}$, i.e., 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. Therefore, the input is of size nk integers for a k-regular digraph of n vertices (that is where each vertex is of in- and out-degree k). Actually, for k-regular digraphs, the list can be replaced by a $n \times k$ table so that the access to any neighbor of any node can be done in constant time. Moreover, we can assume, w.l.g., that the out-neighbor of x connected by an arc of color c_i, $i \in \{1, \ldots, k\}$, is at the same position in the table for all x, i.e., position i in $\mathcal{L}[x]$. (If not yet ordered, such a reordering can be done in $O(nk)$ time.)

Remark. We 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.

Let x_0 be an arbitrary vertex of a 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 occurs 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.g., that x_0 is the zero of the underlying group. Let $\mathcal{T}$ be any breadth-first search spanning tree of G, rooted at x_0, i.e., the path in $\mathcal{T}$ from x_0 to any vertex x is a shortest path in G. We can label each vertex x of G according to the path from x_0 to x in $\mathcal{T}$. More precisely x_0 is labeled $\ell(x_0) = \epsilon = ()$, and every vertex $x \neq x_0$ is labeled $\ell(x) = (\ell(y)|c)$ where y is the father of x in $\mathcal{T}$, c is the color of the arc (y, x), and "$|$" denotes the concatenation of two lists. Once this labeling is fixed, one can define the *orbit* of a vertex $x \neq x_0$ as the ordered sequence $y_0, y_1, y_2, \ldots$ 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 $y_1 = x$. One can then define the *order* of a vertex $x \neq x_0$ as the smallest integer $r > 0$ for which $y_r = x_0$. The order of x is denoted by $\mathbf{o}(x)$, and $\mathbf{o}(x_0)$ is set to 1.

By extension, one can define the order of a color c, denoted by $\mathbf{o}(c)$ 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 a 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. The orbit of a vertex x (resp. of a color c) *passing through* $y \in V$ is defined as the sequence $y_0, y_1, y_2, \ldots$ of vertices where $y_0 = y$ and, for $i > 0$, y_i is the vertex reached from y_{i-1} by following the sequence of colored arcs given by $\ell(x)$ (resp. by following the arc colored c). One can also 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 then simply the sequence $y, y + x, y + 2x, \ldots$

Computing the order of a color c can be easily done in $O(\mathbf{o}(c))$ time by simply following the path of color c from x_0, because adding a color c to a vertex y takes $O(1)$ time (one just have to look for the vertex at the c-th position in the table of the out-neighbors of y to read the result). However, computing the order of an arbitrary vertex x looks harder since following the path $\ell(x)$ from a vertex y requires $\ell(x)$ accesses to the adjacency table. Nevertheless, this complexity can be reduced if we restrict our study to the case $\ell(x) = cc \ldots c = \alpha c$, $\alpha \geq 1$. Of course, one can compute $y + \alpha c$ in $O(\alpha)$ time, but one can do better as shown by the two following lemmas:

Lemma 2 *Let G be a naturally colored Cayley digraph (Γ, S) of order n given by its adjacency table. There is a data structure that can be constructed in $O(nk)$ time, $k = |S|$, which allows to compute the vertex $z = y + \alpha s$ in constant time, for any $y \in V$, any $\alpha \geq 1$ and any $s \in S$.*

Proof. First, let us describe the structure, and show how to compute $z = y + \alpha s$ in constant time using that structure. Then we will show how this structure can be constructed in $O(nk)$ time. For every $s \in S$, G can be viewed as $n_s = n/\mathbf{o}(s)$ s-colored orbits of length $\mathbf{o}(s)$ with arcs of the other colors between the orbits. Let us concentrate on the n_s orbits of s. Order arbitrarily these orbits from 1 to n_s, and, inside each orbit, order the vertices of this orbit from 0 to $\mathbf{o}(s) - 1$ 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_s, one for each element $s \in S$, of respective size $n_s \times \mathbf{o}(s)$. $T_s[i, j]$ is the vertex x at the j-th position in the i-th orbit of s. Conversely, $T[x, s]$ returns the pair (i, j). Computing $z = y + \alpha s$ can be done in constant time using the table T and the tables T_s, $s \in S$, as follows:

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

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

Given a properly colored k-regular digraph G, one can apply the construction given in the proof of Lemma 2, using the set of colors instead of the generating set S. 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 of the proof of Lemma 2 will succeed. The resulting tables are called the *correspondence tables* of the colored digraph G. We clearly have:

Lemma 3 *There is an $O(nk)$-time algorithm which, given a properly colored k-regular digraph G, returns its correspondence tables if exist, and stop otherwise.*

Proof. Since G is properly colored, the only thing that must be checked is that all orbits have the same length. This can be done on the fly, and with no additional cost during the construction of the correspondence tables presented in Lemma 2. ■

Notation. We will denote by $\mathbf{o}(x)$ the *order* of x, whatever x is a color, a vertex of a digraph, or an element of a group. Similarly, we will denote by $\langle x_1, \ldots, x_s\rangle$ the *orbit* of $x_1, \ldots, x_s$, that is the set of the linear combinations of $x_1, \ldots, x_s$, whatever the x_i's are vertices of a digraph, or elements of a group.

3 Recognizing colored Cayley digraphs on abelian groups

An abelian group Γ of order $n = \Pi_{i=1}^{t} p_i^{m_i}$ 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

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

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

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

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 $\mathbb{Z}_{n_1} \times \ldots \times \mathbb{Z}_{n_r}$, $n_i \geq n_{i+1}$. (Note that n_{i+1} divides n_i.)

Our recognition algorithm performs in three phases. The first phase consists to find 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 of our algorithm, to label the vertices, and to compute a candidate for S such that $G = (\Gamma, S)$. Finally, the third phase of our algorithm consists to check whether or not G is color-isomorphic to (Γ, S). 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 Equation 1. It will yield labels of $\sum_{i=1}^{t} r_i$ coordinates. We will refine this construction later, to get labels of r coordinates.

Let us start by a simple observation.

Observation 1 Let $G = (\mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}}, \{s_1, \ldots, s_k\})$, where $k \geq 2$, p and q are prime integers, $p \neq q$, $\alpha_1 \geq \alpha_2$, and $\beta_1 \geq \beta_2$. It may happen that none of the s_i's is of order p^{α_1}. However, one of the s_i's is of order ap^{α_1}, $a = q^{\gamma}$, $\gamma \geq 1$. Indeed, otherwise, the elements of order p^{α_1} could not be obtained by linear combination of the s_i's. Assume, w.l.g., that $\mathbf{o}(s_1) = ap^{\alpha_1}$. Then set $g_1 = as_1$. If $s_1 = (x_1, x_2, x_3, x_4) \in \mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}}$. We have

$$g_1 = as_1 = (x_1', x_2', 0, 0) \in \mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}}$$

with $\mathbf{o}(x_1') = \mathbf{o}(g_1) = p^{\alpha_1}$. Let $\widehat{\Gamma} = (\mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}})/\langle g_1\rangle \simeq \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}}$. Since $\{\widehat{s_1}, \ldots, \widehat{s_k}\}$ generates $\widehat{\Gamma}$, one of these elements has order bp^{α_2} with $b = q^{\gamma'}$, $\gamma' \geq 0$. Actually, this element cannot be s_1 because s_1 has order a in $\widehat{\Gamma}$. Assume, w.l.g., that $\mathbf{o}(\widehat{s_2}) = bp^{\alpha_2}$, $b = q^{\gamma'}$, $\gamma' \geq 0$. Let us show that there exists $u \in \{0, \ldots, p^{\alpha_2} - 1\}$ such that $\mathbf{o}(ug_1 + bs_2) = p^{\alpha_2}$, and $\langle g_1\rangle \cap \langle ug_1 + bs_2\rangle = \{0\}$. Let λ such that $\lambda x_1' = 1$. We have $\lambda g_1 = (1, x_2'', 0, 0)$. Assuming that $s_2 = (y_1, y_2, y_3, y_4)$, we get $bs_2 = (y_1', y_2', 0, 0)$ because $\mathbf{o}(\widehat{s_2}) = q^{\gamma'}p^{\alpha_2}$ and $g_1 = (x_1', x_2', 0, 0)$. Thus, there exists u such that $g_2 = ug_1 + bs_2 = (0, y_2'', 0, 0)$. We have $g_2 \in \widehat{bs_2}$, and thus $\mathbf{o}(g_2) \geq p^{\alpha_2}$. On the other hand $\mathbf{o}((0, y_2'', 0, 0)) \leq p^{\alpha_2}$. Therefore $\mathbf{o}(g_2) = p^{\alpha_2}$. Finally, $\mu g_1 \in \langle g_2 \rangle$ implies $\mu x_1' = 0$, that is $\mu = p^{\alpha_1}$, and thus $\mu(x_1', x_2', 0, 0) = (0, 0, 0, 0)$. Therefore $\langle g_1 \rangle \cap \langle g_2 \rangle = \{0\}$. To summarize,

$$g_2 = ug_1 + bs_2 = (0, y_2'', 0, 0) \in \mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}}$$

with $\mathbf{o}(y_2'') = \mathbf{o}(g_2) = p^{\alpha_2}$.

The β_i's can be used in the same way to derive g_3 and g_4 satisfying

$$g_3 = cs_{i_1} = (0, 0, z_3, z_4) \in \mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}}$$

with $\mathbf{o}(z_3) = \mathbf{o}(g_3) = q^{\beta_1}$. Here $\mathbf{o}(s_{i_1}) = cq^{\beta_1}$ where $c = p^{\delta}$, $\delta \geq 0$. Note that s_{i_1} can be any of the s_i, in particular, it can be s_1 or s_2. From g_3, one can get

$$g_4 = u'g_3 + ds_{i_2} = (0, 0, 0, w_4) \in \mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}}$$

with $\mathbf{o}(w_4) = \mathbf{o}(g_4) = p^{\beta_2}$. Again, $\mathbf{o}(\widehat{s_{i_2}}) = dq^{\beta_2}$, $d = p^{\delta'}$, $\delta' \geq 0$, where $\widehat{s_{i_2}}$ denotes the class of s_{i_2} in $\widehat{\Gamma}' = (\mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_1}} \times \mathbb{Z}_{q^{\beta_2}})/\langle g_3 \rangle \simeq \mathbb{Z}_{p^{\alpha_1}} \times \mathbb{Z}_{p^{\alpha_2}} \times \mathbb{Z}_{q^{\beta_2}}$.

The conclusion of this observation is that the system of the s_i's can be triangulated by block to get a basis of the $\mathbb{Z}$-module Γ. Intuitively, for every prime p_i in the decomposition of n, we will get a basis $g_{i,1}, \ldots, g_{i,r_i}$ of the subgroup of Γ isomorphic to $\mathbb{Z}_{p_i^{\alpha_{i,1}}} \times \ldots \times \mathbb{Z}_{p_i^{\alpha_{i,r_i}}}$. In the same way as $\langle g_1, g_2 \rangle \cap \langle g_3, g_4 \rangle = \{0\}$, the subgroup generated by the $g_{i,j}$'s of one prime p_i will intersect the subgroup generated by the $g_{i',j}$'s of another prime $p_{i'}$ only once. This is formally stated in the following lemma.

Lemma 4 *Let $G = (\Gamma, \{s_1, \ldots, s_k\})$ be an abelian Cayley digraph. Assume that the unknown group Γ is described by Equation 1. Then, one can compute in $O(nk)$ time $\sum_{i=1}^{t} r_i$ vertices $g_{i,j}$, $i \in \{1, \ldots, t\}$, $j \in \{1, \ldots, r_i\}$, such that $\mathbf{o}(g_{i,j}) = p_i^{\alpha_{i,j}}$ for every i and j, and*

- *$\langle g_{i,j} \rangle \cap \langle g_{i,1}, \ldots, g_{i,j-1} \rangle = \{0\}$ for every i, and every $j > 1$;*
- *$\langle g_{i,1}, \ldots, g_{i,r_i} \rangle \cap \langle g_{i',1}, \ldots, g_{i',r_{i'}} \rangle = \{0\}$ for every $i \neq i'$.*

Proof. The construction is based on Observation 1. Let us start by p_1. One of the s_i's has to be of order $a_i p_1^{\alpha_{1,1}}$ where p_1 does not divide a_i. Assume, w.l.g., that s_1 is such that the multiplicity of p_1 in the prime decomposition of $\mathbf{o}(s_1)$ is maximum. Then $\mathbf{o}(s_1) = a_1 p_1^{\alpha_{1,1}}$, and thus we set $g_{1,1} = a_1 s_1$ so that $\mathbf{o}(g_{1,1}) = p_1^{\alpha_{1,1}}$.

Assume that we have already computed $g_{1,1}, \ldots, g_{1,i-1}$ and $\langle g_{1,1}, \ldots, g_{1,i-1} \rangle$. We can assume, w.l.g., that for every j, $1 \leq j \leq i-1$, $g_{1,j} \in \langle a_1 s_1, \ldots, a_j s_j \rangle$. Let us compute $\alpha_{1,i}$. For every $j \geq i$, let $\mathbf{o}(s_j) = a_j p_1^{\gamma_j}$ where p_1 does not divide a_j. Let β_j be the smallest positive integer such that $\beta_j a_j s_j \in \langle g_{1,1}, \ldots, g_{1,i-1} \rangle$. β_j is actually the order of $a_j s_j$ in the quotient of $\Gamma / \langle g_{1,1}, \ldots, g_{1,i-1} \rangle$. Therefore, $\alpha_{1,i} = \max_j \beta_j$. Assume, w.l.g., that $\alpha_{1,i} = \beta_i$. Therefore, by similar arguments as Observation 1, there exists a sequence $u_1, \ldots, u_{i-1}$, $0 \leq u_j \leq p_1^{\alpha_{1,j}} - 1$ for $1 \leq j < i$, such that $g_{1,i} = u_1 a_1 s_1 + \ldots + u_{i-1} a_{i-1} s_{i-1} + a_i s_i$ makes the sequence $g_{1,1}, \ldots, g_{1,i-1}, g_{1,i}$ to satisfy the condition of the lemma.

To compute $u_1, \ldots, u_{i-1}$, we compute the orbits (actually, only partial orbits, that is the $p_1^{\alpha_{1,i}}$ first vertices of the orbits) of $u_1 a_1 s_1 + \ldots + u_{i-1} a_{i-1} s_{i-1}$, $0 \leq u_j \leq p_1^{\alpha_{1,j}} - 1$ for $1 \leq j < i$. There are $p_1^{\alpha_{1,1}} \ldots p_1^{\alpha_{1,i-1}}$ orbits, and the partial orbits of $u_1 a_1 s_1 + \ldots + u_{i-2} a_{i-2} s_{i-2}$ of length $p_1^{\alpha_{1,i-1}} \geq p_1^{\alpha_{1,i}}$ have been computed before, when looking for $g_{1,i-1}$. By application of Lemma 2, the partial orbits of all the $u_1a_1s_1+\ldots+u_{i-1}a_{i-1}s_{i-1}$'s can be computed in $O((p_1^{\alpha_{1,1}}\ldots p_1^{\alpha_{1,i-1}})p_1^{\alpha_{1,i}})$ time. Based on these orbits, and applying again Lemma 2, one can compute $\langle u_1a_1s_1+\ldots+u_{i-1}a_{i-1}s_{i-1}+a_is_i\rangle$, in $O((p_1^{\alpha_{1,1}}\ldots p_1^{\alpha_{1,i-1}})p_1^{\alpha_{1,i}})$ time (again, only partial orbits of length $p_1^{\alpha_{1,i}}$ are constructed). At least one of these orbits satisfies the condition of the statement of this lemma. Let $g_{1,i}=u_1a_1s_1+\ldots+u_{i-1}a_{i-1}s_{i-1}+a_is_i$ with the corresponding coefficients. We have now $\langle g_{1,i}\rangle\cap\langle g_{1,1},\ldots,g_{i-1,1}\rangle=\{0\}$ and $\mathbf{o}(g_{1,i})=p_1^{\alpha_{1,i}}$, and it just remains to compute $\langle g_{1,1},\ldots,g_{1,i}\rangle$. The computation of the $g_{1,i}$'s is completed when $\sum_i\alpha_{1,i}=m_1$ where m_1 is the multiplicity of p_1 in n.

The computation of the $g_{i,j}$'s for $i>1$ is performed in the same way (of course the order in which the generators appear in the setting of the $g_{i,j}$'s may differ).

The complexity of the resulting algorithm is the sum over all the t primes p_i of the cost of computing the $g_{i,j}$'s, $i=1,\ldots,t$. The cost of computing $g_{i,j}$ for a given pair (i,j) is in turn the sum of (1) the cost of finding the corresponding generator, say s_j, (2) the cost of computing $\langle u_1s_1+\ldots+u_{j-1}s_{j-1}+a_js_j\rangle$, and (3) the cost of computing $\langle g_{i,1},\ldots,g_{i,j}\rangle$. The complexity of (1) is $(k-j)p_i^{\alpha_{i,j}}$. The complexity of (2) and (3) is $O((p_i^{\alpha_{i,1}}\ldots p_i^{\alpha_{i,j-1}})p_i^{\alpha_{i,j}})$. we get a complexity of

$$\sum_{i=1}^{t}(\sum_{j=2}^{r_i}O(kp^{\alpha_{i,j}})+\sum_{j=1}^{r_i}O((p_i^{\alpha_{i,1}}\ldots p_i^{\alpha_{i,j-1}})p_i^{\alpha_{i,j}}))=O(k\sqrt{n}+n)$$

not including the $O(nk)$ time to compute the correspondence tables. Note that all the prime decompositions appearing in the algorithm can be done in at most $k\log n$ time. The total complexity of the algorithm is thus $O(nk)$. ■

We are now ready to prove the main result of this section.

Theorem 1 *There is an $O(m\min\{k,\log n\})$-time algorithm which, given a properly colored k-regular digraph G of n vertices and m arcs, returns 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.*

Proof. 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 $\sum_i r_i$ 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 $\Gamma=\mathbb{Z}_{n_1}\times\ldots\times\mathbb{Z}_{n_r}$ as follows.

Node x_0 is labeled by the r-tuple $(0,0,\ldots,0)$. The β-th vertex of the orbit of h_1 is labeled the r-tuple $(\beta,0,\ldots,0)$. The γ-th vertex of the orbit of h_2 passing through $(\beta,0,\ldots,0)$ (starting from $(\beta,0,\ldots,0)$) is labeled $(\beta,\gamma,0,\ldots,0)$. And so on. Thanks to Lemma 2, 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.

Once the labeling has been performed, it just 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

$$\left\{\begin{array}{lcll} y_1 - x_1 & = & \psi_1(c_i) & (\mathrm{mod}\ n_1); \\ y_2 - x_2 & = & \psi_2(c_i) & (\mathrm{mod}\ n_2); \\ & \vdots & & \\ y_r - x_r & = & \psi_r(c_i) & (\mathrm{mod}\ n_r); \end{array}\right. \tag{2}$$

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. Indeed, the map $\phi : \Gamma \mapsto \Gamma$ such that $\phi(a_1 h_1 + \ldots + a_r h_r) = (a_1, \ldots, a_r)$ 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 $\psi(c_i) = \phi(s_i)$. This checking takes a time $O(rm)$ because every label is in $\mathbb{Z}^r$ and $r \leq \min\{k, \log n\}$. ■

4 Recognizing colored abelian Cayley digraphs of rank r

The following result is actually a simple corollary of Theorem 1.

Corollary 1 *There is an $O(mr)$-time algorithm which, given a properly colored regular digraph G of m arcs, returns 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.*

Proof. The algorithm is a simple variant of the algorithm described in the previous section for arbitrary abelian group. Once the $g_{i,j}$'s have been computed (Lemma 4) in $O(m)$ time, 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. If it is equal to r, the algorithm will perform in $O(mr)$ time as computed in the proof of Theorem 1. ■

4.1 A specific case: circulant digraphs

The case $r = 1$ deserves a specific attention as it corresponds to the class of circulant digraphs. Thus, we have designed a particular and elegant algorithm for that case. This algorithm performs optimally in $\Theta(m)$ time. It is based on the following basic property:

Lemma 5 *Let a and b be two elements of $\mathbb{Z}_n$ satisfying $\mathbf{o}(a) < n$, and $b \notin \langle a \rangle$. Let β be the smallest positive integer such that $\beta b = \gamma a$ for some $\gamma \in \{1, \ldots, \mathbf{o}(a)\}$. Let d defined by $d = ua + vb$ where (u, v) is the solution of $u\beta + v\gamma = \gcd(\beta, \gamma)$ returned by the Euclid's algorithm if $\gcd(\beta, \gamma) \neq \beta$, and $u = 1 - \frac{\gamma}{\beta}$, $v = 1$ otherwise. Then $\mathbf{o}(d) = \lambda\, \mathbf{o}(a)$ with $\lambda \geq 2$.*

Proof. Let $\delta = \gcd(\beta, \gamma)$. Note that $\beta d = \delta a$ because

$$\beta d = \beta(ua + vb) = u\beta a + v\gamma a = (u\beta + v\gamma)a.$$

Similarly, $\gamma d = \delta b$. Let $\delta' = \gcd(\mathbf{o}(a), \delta)$. We have $\delta' | \mathbf{o}(a)$, $\delta' | \beta$, and $\delta' | \gamma$. Thus $\mathbf{o}(\frac{\beta}{\delta'} b) | \mathbf{o}(a)$ because

$$\mathbf{o}(a) \frac{\beta}{\delta'} b = \frac{\mathbf{o}(a)}{\delta'} \beta b = \frac{\mathbf{o}(a)}{\delta'} \gamma a = \frac{\gamma}{\delta'} \mathbf{o}(a) a = 0.$$

Therefore, $\frac{\beta}{\delta'} b \in \langle a \rangle$, and hence $\delta' = 1$, by definition of β. Since $\delta a = \beta d$ and $\gcd(\mathbf{o}(a), \delta) = 1$, we get that βd generates $\langle a \rangle$, and therefore $a \in \langle d \rangle$, that is $\mathbf{o}(d) = \lambda\, \mathbf{o}(a)$ for some $\lambda \geq 1$. To prove $\lambda \geq 2$, we consider two cases.

Case 1: $\frac{\beta}{\delta} \neq 1$. Then $\delta b \notin \langle a \rangle$, and thus for any μ, $0 \leq \mu < \mathbf{o}(a)$, $\mu a + \delta b \in \langle d \rangle$, i.e., $\langle a \rangle + \delta b \in \langle d \rangle$. Since we already know that $a \in \langle d \rangle$, since $\langle a \rangle$ and $\langle a \rangle + \delta b$ contain $\mathbf{o}(a)$ elements each, and since $\langle a \rangle \cap (\langle a \rangle + \delta b) = \varnothing$, we get $\mathbf{o}(d) \geq 2\,\mathbf{o}(a)$.
Case 2: $\frac{\beta}{\delta} = 1$. Then $d = (1 - \frac{\gamma}{\beta})a + b$, and since $a \in \langle d \rangle$, there is an integer α such that $\alpha d = a$. Therefore $b = d(\alpha(\frac{\gamma}{\beta} - 1) + 1)$, and hence $b \in \langle d \rangle$ and $\mathbf{o}(d) = \beta\,\mathbf{o}(a)$. The proof is completed by noticing that $\beta \geq 2$ since $b \notin \langle a \rangle$. ■

Note that it is often the case that $\langle d \rangle = \langle a, b \rangle$, and the group $\langle a, b \rangle$ contains exactly $\beta\,\mathbf{o}(a)$ elements. In particular, if $\gcd(\beta, \gamma) = 1$ then we have $\gamma d = b$ and $\beta d = a$, which implies that $\langle d \rangle = \langle a, b \rangle$, that is $\mathbf{o}(d) = \beta\,\mathbf{o}(a)$. However, there are cases in which $\gcd(\beta, \gamma) \neq 1$, e.g., in $\mathbb{Z}_{60}$, $a = 50$ and $b = 3$ yield $\mathbf{o}(a) = 15$, $\beta = 4$, and $\gamma = 6$, with $\gcd(\beta, \gamma) = 2$. Actually, since $a \in \langle d \rangle$, and $d = ua + vb$, $\mathbf{o}(d) = \beta\,\mathbf{o}(a)$ if and only if $\gcd(v, \mathbf{o}(b)) = 1$. By computer search, we were able to determine that the smallest value of n for which there exist a and b such that $\mathbf{o}(d) < \beta\,\mathbf{o}(a)$ is $n = 150$. In $\mathbb{Z}_{150}$, $a = 10$ and $b = 51$ yield $\mathbf{o}(a) = 15$, $\mathbf{o}(b) = 50$, $\beta = 10$, $\gamma = 6$, $\gcd(\beta, \gamma) = 2$, $u = -1$, $v = 2$, and $d = 92$. We get $\mathbf{o}(d) = 75 < \beta\,\mathbf{o}(a) = 150$. Anyway, for the purpose of our algorithm, it is sufficient that $\mathbf{o}(d) \geq 2\,\mathbf{o}(a) = 30$.

Based on Lemma 5, we have derived the algorithm *Recognize circulant* which takes as input a properly colored k-regular colored digraph G of n vertices, described by its adjacency table, and tells whether or not G is color-isomorphic to a naturally colored circulant digraph, and, in the affirmative, returns the set of chords. (In the description of the algorithm, $\mathcal{T}$ and x_0 refers to the notation of Section 2.)

Algorithm *Recognize circulant***:**

1. Pick any out-neighbor a of x_0, and mark it as "chosen"; If $\mathbf{o}(a) = n$ then go directly to 6;
2. Pick any not yet chosen out-neighbor b of any already chosen vertex, and mark it as "chosen";
3. Compute β defined as the smallest positive integer such that $\beta b = \gamma a$, for some γ;
4. Let u and v as defined in the statement of Lemma 5, and let $d = ua + vb$; Compute $\langle d \rangle$;
5. Case:
    1. $\mathbf{o}(d) = n$ then $a \leftarrow d$ and go to 6;
    2. $2\,\mathbf{o}(a) \leq \mathbf{o}(d) < n$ then $a \leftarrow d$ and go to 2;
    3. $\mathbf{o}(d) < 2\,\mathbf{o}(a)$ then return "no" and stop;
6. Label the vertices from 0 to $n-1$ according to the sequence defined by the orbit of a, x_0 being labeled 0; The label of x is denoted $\nu(x)$;
7. Check that for every color c, and any two arcs (x, y) and (x', y') of color c,

$$\nu(y) - \nu(x) \equiv \nu(y') - \nu(x') \pmod{n}.$$

Theorem 2 *The algorithm Recognize circulant checks in $O(nk) = O(m)$ time whether a k-regular colored digraph G of n vertices and m arcss is color-isomorphic to a naturally colored circulant digraph.*

Proof. Algorithm *Recognize circulant* performs in phases, each phase consisting of Instructions 2 to 5. (Note that there are at most $\lceil \log_2 n \rceil$ phases due to Instruction 5.2 motivated by Lemma 5.)

The aim of these phases is to find a vertex of order n since, if the input digraph is a Cayley digraph on $\mathbb{Z}_n$, such a vertex must exist. If this search fails (Instruction 5.3), then the input digraph cannot be a Cayley digraph on $\mathbb{Z}_n$, and the algorithm stops. Once a vertex a of order n has been found, the algorithm labels the vertices using that vertex (Instruction 6) since, if the input digraph is a Cayley digraph on $\mathbb{Z}_n$, then, by Ádám-isomorphism, a can be assumed to be 1. From Lemma 1, once the labeling has been performed, the Instruction 7 checks that the arcs of the same color correspond to a unique chord.

The rest of the proof is concerned with the cost of the Algorithm *Recognize circulant*. The algorithm construct a sequence $a_0, a_1, a_2, \dots$ of vertices, where a_0 is the vertex a of Instruction 1, and a_i being the current vertex, a_{i+1} is the vertex d of Instruction 5.2. This sequence satisfies $\mathbf{o}(a_{i+1}) \geq 2\,\mathbf{o}(a_i)$.

Let us compute the cost of the i-th phase, $i \geq 1$. It is important to note that, in Instruction 2, the vertex b is picked among the vertices that are out-neighbors of previously chosen vertices. In other words, one can assume w.l.g, that b's parent in $\mathcal{T}$ has been considered at a previous phase (where $\mathcal{T}$ is the breadth-first search spanning tree described in Section 2). Let us denote b' the parent of b in $\mathcal{T}$, and assume (b', b) is colored c. For any $\beta \geq 1$, $\beta b = \beta(b' + c) = \beta b' + \beta c = y + \beta c$ where y has been identified at a previous phase (when constructing $\langle b' \rangle$). From Lemma 2, the operation $y + \beta c$ can be performed in constant time as soon as the correspondence tables have been set. This setting takes $O(nk)$ time, but, once it has been done, Phase i has a cost $O(\mathbf{o}(a_i))$.

Therefore, identifying a vertex of order n takes $\sum_{i=0}^{p} O(\mathbf{o}(a_i))$ time, for p phases. Since, $\mathbf{o}(a_p) = n$ and $\mathbf{o}(a_{i+1}) \geq 2\,\mathbf{o}(a_i)$, identifying a vertex of order n takes no more than $O(n)$ time in total.

The cost of Instruction 6 is $O(\mathbf{o}(a_p)) = O(n)$ (again, assuming that the correspondence tables have been set), and, finally, Instruction 7 has a cost of $O(nk)$ because there are nk arcs, and each arc can be treated in constant time.

In conclusion, we have $O(nk)$ for the setting of the correspondence tables, $O(n)$ for finding a vertex of order n, $O(n)$ for labeling the vertices, and $O(nk)$ for checking the correctness of the coloring. Thus Algorithm *Recognize circulant* has a total cost of $O(nk)$. ■

4.2 Graph color-isomorphism

Note that if the input digraph G of algorithm *Recognize circulant* is a circulant digraph, then the algorithm *Recognize circulant* will not only answer positively to the question whether G is color-isomorphic to a naturally colored circulant digraph, but it will also return a digraph G' which is Ádám-isomorphic to G.

Corollary 2 *Let $\mathcal{C}$ be the class of colored digraphs that are color-isomorphic to a naturally colored circulant digraph. There is an $O(m)$-time algorithm that determines, for any pair G and H of digraphs in $\mathcal{C}$, both of size m, whether G is color-isomorphic to H.*

Proof. Let G and H be the two input digraphs, both of order n. Let S and T be the set of chords returned by Algorithm *Recognize circulant* on G and H respectively. From Lemma 1, 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 $S = \{s_1, \dots, s_k\}$, and $T = \{t_1, \dots, t_k\}$. There are at most k candidates for λ: the solutions of $\lambda s_1 = t_i$, $i = 1, \dots, k$, if exists. Each candidate can be computed in $O(\log n)$ time by the Euclid's algorithm. For every candidate, one checks whether $\lambda S = T$. All these operations take a negligible time in front of the $O(nk) = O(m)$ time required by the algorithm *Recognize circulant*. ■

Remark. The previous result can be generalized to abelian Cayley digraphs by noticing, as a generalization of Lemma 1, that 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 $\phi(S) = T$, 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 1 to compute their respective groups Γ_1 and Γ_2, and generating sets S and T. If $\Gamma_1 \neq \Gamma_2$, then G_1 and G_2 are not color-isomorphic. Otherwise, one can test the $k!$ one-to-one functions mapping S to T. Let $B = \{e_i, i = 1, \ldots, r\}$, $e_i = (0, \ldots, 0, 1, 0, \ldots, 0)$ where the 1 is at the i-th position. Every test takes $O(kr^2)$ time to compute $\phi(B)$, plus $O(nk)$ time to check whether $\phi(B)$ is a basis. Therefore, if $k = O(1)$, one can check the color-isomorphism between two naturally colored abelian Cayley digraphs in $O(n)$ time. For digraphs of large degree, the $k!$ factor makes this algorithm non practical.

Acknowledgments. The authors are thankful to Nicola Santoro who suggested them to study the color-isomorphism problem. The authors also want to express their deep thanks to Charles Delorme who gave them many hints 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. Grotschel, and L. Lovász, editors, *Handbook of Combinatorics*, volume 2. Elsevier and MIT Press, 1995.

[4] G. Birkhoff and S. Mac Lane. *Algebra.* Macmillan, 1967.

[5] P. Boldi and S. Vigna. On the complexity of deciding sense of direction. In *Colloquium on Structural Information and Communication Complexity (SIROCCO)*, volume 2 of *Int. Informatics Series*, pages 39–51. Carleton University Press, 1995.

[6] P. Boldi and S. Vigna. Minimal sense of direction and decision problems for Cayley graphs. *Information Processing Letters*, 64:299–303, 1997.

[7] C. Delorme, O. Favaron, and M. Maheo. Isomorphisms of Cayley multigraphs of degree 4 on finite abelian groups. *European Journal of Combinatorics*, 13:59–61, 1992.

[8] B. Elspas and J. Turner. Graphs with circulant adjacency matrices. *J. Comb. Theory*, 9:229–240, 1970.

[9] 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.

[10] C. Li, C. Praeger, and M. Xu. On finite groups with the Cayley isomorphism property. *Journal of Graph Theory*, 27:21–31, 1998.

[11] G. Miller. On the $n^{\log n}$ isomorphism technique. In *Proceedings Tenth Annual ACM Symposium on Theory of Computing (STOC)*, pages 51–58, 1978.

[12] J. Morris. Isomorphic Cayley graphs on nonisomorphic groups. *Journal of Graph Theory*, 31:345–362, 1999.

[13] Muzychuk. On Ádám's conjecture for circulant graphs. *Discrete Mathematics*, 167/168:497–510, 1997. Erratum: ibid., 176, 1997.

[14] M. Muzychuk and G. Tinhoffer. Recognizing circulant graphs of prime order in polynomial time. *The electronic journal of combinatorics*, 3, 1998.

[15] S. Toida. A note on Ádám's conjecture. *Journal of Combinatorial Theory*, B 23:247–250, 1977.

[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.

A Proof of Lemma 1

Let $G_1 = (\mathbb{Z}_n, S)$ and $G_2 = (\mathbb{Z}_n, T)$ be two naturally colored circulant digraphs, with $S = \{s_1, \ldots, s_k\}$ and $T = \{t_1, \ldots, t_k\}$.

Let first assume that G_1 and G_2 are Ádám-isomorphic, and hence let γ be the invertible element of $\mathbb{Z}_n$ such that $T = \gamma \cdot S$. The mapping $\Phi : \mathbb{Z}_n \mapsto \mathbb{Z}_n$ defined by $\Phi(v) = \gamma v$, and the function $\psi : S \mapsto T$ that maps s_i to γs_i make G_1 and G_2 color-isomorphic.

Conversely, assume that G_1 and G_2 are color-isomorphic. The coloring of G_1 and G_2 are naturally given by the labeling of the elements of S and T from 1 to k. We can assume, w.l.g., that the function ψ defining the correspondence between the colors is the identity of $\{1, \ldots, k\}$. Also, due to the vertex-symmetry of both G_1 and G_2, we can assume, w.l.g., that the function $\Phi : \mathbb{Z}_n \mapsto \mathbb{Z}_n$ defining the isomorphism between G_1 and G_2 satisfies $\Phi(0) = 0$. For every $x, y \in \mathbb{Z}_n$, (x, y) is an arc of color c of G_1 if and only if $(\Phi(x), \Phi(y))$ is an arc of color $\psi(c)$, that is of color c, of G_2. Thus, since $\Phi(0) = 0$, $\Phi(s_i) = t_i$ for every $i \in \{1, \ldots, k\}$. Now, let $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ be a sequence of generators of G_1 such that $\sum_{j=1}^{r} s_{i_j} = 1$. (Note that the same generator may appear several times in this sequence.) For every vertex v of G_1, the sequence of arcs $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ starting at v is a path from v to $v+1$. On the other hand, $t_{i_1}, t_{i_2}, \ldots, t_{i_r}$ is a sequence of generators of G_2 satisfying $\sum_{j=1}^{r} t_{i_j} = \Phi(1)$, and thus, for every vertex w of G_2, the sequence of arcs $t_{i_1}, t_{i_2}, \ldots, t_{i_r}$ starting at w is a path from w to $w + \Phi(1)$. Moreover, since Φ is an isomorphism, the order of the elements is preserved, and hence $\Phi(1)$ is also of order n, that is $\Phi(1)$ is invertible. For every $i \in \{1, \ldots, k\}$, the sequence $s_{i_1}, s_{i_2}, \ldots, s_{i_r}$ repeated s_i times yields a path from 0 to s_i in G_1. Similarly, the sequence $t_{i_1}, t_{i_2}, \ldots, t_{i_r}$ repeated s_i times yields a path from 0 to $s_i\Phi(1)$ in G_2. Now observe that the sequence from 0 to s_i in G_1, and the sequence from 0 to $s_i\Phi(1)$ in G_2, have exactly the same coloring. Therefore, $\Phi(s_i) = s_i\Phi(1)$. Since $t_i = \Phi(s_i)$, we get $t_i = \Phi(1)s_i$ for every $i \in \{1, \ldots, k\}$. Therefore $T = \Phi(1) \cdot S$, with $\Phi(1)$ invertible, and hence G_1 and G_2 are Ádám-isomorphic. ■