Graphs and Combinatorics (2024) 40:120
https://doi.org/10.1007/s00373-024-02852-y

ORIGINAL PAPER

Finding k Shortest Paths in Cayley Graphs of Finite Groups

Dohan Kim[1]

Received: 6 October 2023 / Revised: 16 October 2024 / Accepted: 23 October 2024

Abstract
We present a new method for finding k shortest paths between any two vertices in the Cayley graph $\mathrm{Cay}(G, S)$ of a finite group G with its generating set S closed under inverses. By using a reduced convergent rewriting system R for G, we first find the lexicographically minimal shortest path between two vertices in $\mathrm{Cay}(G, S)$. Then, by symmetrizing the length-preserving rules of R, we provide a polynomial time algorithm (in the size of certain rewrite rules, the lexicographically minimal shortest path, and k) for finding k shortest paths between two vertices in $\mathrm{Cay}(G, S)$. Our implementation of finding k shortest paths between two vertices in $\mathrm{Cay}(G, S)$ is also discussed.

Keywords k shortest path · Cayley graph · Finite group · String rewriting system

Mathematics Subject Classification 05C25 · 05C30 · 05A05 · 16S15

1 Introduction

Finding a shortest path between two vertices in a graph plays a fundamental role in graph theory [16, 31]. The traditional algorithms for finding a shortest path between two vertices in a graph often assume that the graph is weighted [31]. For unweighted graphs such as Cayley graphs discussed in this paper, those algorithms often rely on the breadth-first search algorithm which has some scalability issues for Cayley graphs [10, 11, 32]. (For example, the Cayley graph $\mathrm{Cay}(\mathfrak{S}_n, S)$ of the symmetric group $\mathfrak{S}_n$ with its generating set S has $O(n!)$ vertices and $O(n!|S|)$ edges.)

Therefore, the traditional approaches to finding shortest paths are not suitable for Cayley graphs of finite groups in general.

This paper first discusses how to find the lexicographically minimal shortest path between two vertices in $\mathrm{Cay}(G, S)$ using a *(reduced) convergent rewriting system* R for G [26] and a shortlex order $\prec$. By using the length-preserving rules of the *preperfect rewriting system* T [6, 19] obtained from the *Thue resolution* [19] of R, we show how

✉ Dohan Kim
dohan.kim@uibk.ac.at

[1] Department of Computer Science, University of Innsbruck, Innsbruck, Austria

Published online: 07 November 2024

to find k-shortest paths[1] (if they exist) between two vertices in $\mathrm{Cay}(G, S)$ starting with the lexicographically minimal shortest path between two vertices in $\mathrm{Cay}(G, S)$.

We also provide our implementation, called *Geodesics*, in order to provide a tool for the k shortest path enumeration algorithm discussed in this paper.

Throughout this paper, we assume that a finite group presentation of G w.r.t. S is given for $\mathrm{Cay}(G, S)$. If $\mathrm{Cay}(G, S)$ is given without a group presentation of G w.r.t. S, one may need to construct a group presentation of G in order to use our rewriting methods. The reader may refer to [9, 12, 36] for constructing a group presentation of a finite group G w.r.t. S using the *fundamental circuits* in $\mathrm{Cay}(G, S)$.

2 Preliminaries

Definitions and results used in this section are found in [3, 6, 17, 19, 23–28, 36]. We assume that the reader has some familiarity with the theory of finite groups.

Let X be a set and X^* be the free monoid over X. Then X^* consists of all words over X, where the empty word is denoted by ε or 1. A (string) *rewriting system* on X^* is a subset $R \subseteq X^* \times X^*$. An ordered pair $(u, v) \in R$ is called a *rewrite rule* (or simply called a *rule*) of R. (We also write $u \to v \in R$ instead of $(u, v) \in R$.) Given a rewriting system R on X^*, write $x \to_R y$ for $x, y \in X^*$ if there exists a rule $(v_1, v_2) \in R$ and words $u, w \in X^*$ such that $x = uv_1w$ and $y = uv_2w$. We write $x \to y$ instead of $x \to_R y$ if R is clear from context.

Let $\overset{*}{\to}$ (resp. $\overset{*}{\leftrightarrow}$) denote the reflexive and transitive (resp. reflexive, transitive, and symmetric) closure of $\to$. Write $x \overset{*}{\to} y$ if $x = y$ or $x \to x_1 \to x_2 \to \cdots \to y$ for some finite chain of arrows $\to$.

Given a monoid M, a rewriting system R on X^* is called a *rewriting system for* M if $< X \mid l = r \text{ if } (l \to r) \in R >$ is a presentation of M. Here, the relation $\overset{*}{\leftrightarrow}_R \subseteq X^* \times X^*$ is a *congruence* on X^*. The congruence class $[w]_R$ of a word $w \in X^*$ is defined as $[w]_R := \{v \in X^* \mid w \overset{*}{\leftrightarrow}_R v\}$. The congruence classes of $\overset{*}{\leftrightarrow}_R$ on X^* form a monoid isomorphic to M.

In the remainder of this paper, if $< X \mid \mathcal{R} >$ is a monoid presentation, then we denote it by $\mathrm{Mon} < X \mid \mathcal{R} >$ in order to differentiate it from a group presentation.

Let X be a set of generators of a finite group G, let $X_\alpha = \{x^\alpha \mid x \in X, \alpha \in \{1, -1\}\}$, and let $\mathcal{R}$ be a set of relations of G. Then the group defined by the presentation $< X \mid \mathcal{R} >$ is equivalent to the monoid defined by the presentation $\mathrm{Mon} < X_\alpha \mid I_X \cup \mathcal{R} >$, where $I_X = \{xx^{-1} = 1 \mid x \in X_\alpha\}$.

Let S be a set of generators of a finite group G as a monoid, which implies that S is closed under inverses. A rewriting system R for a finite group G (on S^*) is a rewriting system for G as a monoid, where the elements of G are congruence classes of $\overset{*}{\leftrightarrow}_R$ on S^*.

One of the most common orderings used in (string) rewriting systems is the *length-plus-lexicographic ordering*, also called the *shortlex ordering*. Let $\prec$ be a shortlex order on S^*. Then $v_1 \cdots v_m \prec w_1 \cdots w_n$, provided either that $m < n$ or that $m = n$ and

[1] In this paper, by k shortest paths between two vertices in $\mathrm{Cay}(G, S)$ we mean k shortest paths with the same minimal distance between two vertices in $\mathrm{Cay}(G, S)$

$v_1 \cdots v_m$ comes before $w_1 \cdots w_m$ lexicographically using a (strict total) lexicographic ordering on S. In this paper, the shortlex ordering is total on S^*.

A (string) rewriting system R for a finite group G (on S^*) is said to be *Noetherian* (or *terminating*) if there is no infinite chain of rewritings $x \to x_1 \to x_2 \to \cdots$ for any word $x \in S^*$.

A rewriting system R for G (on S^*) is said to be *compatible* with a shortlex order $\prec$ on S^* if $r \prec l$ for each rule $(l, r) \in R$. If a rewriting system R for G is compatible with a shortlex order $\prec$, then R is Noetherian.

A rewriting system R for G is said to be *confluent* if whenever $x \overset{*}{\to} y_1$ and $x \overset{*}{\to} y_2$, there is a z such that $y_1 \overset{*}{\to} z$ and $y_2 \overset{*}{\to} z$. A rewriting system R for G is said to be *locally confluent* if whenever $x \to y_1$ and $x \to y_2$, there is a z such that $y_1 \overset{*}{\to} z$ and $y_2 \overset{*}{\to} z$. A rewriting system R for G is said to be *convergent* if R is Noetherian and confluent. A word $w \in S^*$ is in *R-normal form* (or *normal form* w.r.t. R) if there is no possible rewriting (or reduction) $w \to v$. A rewriting system R for G is said to be *reduced* if each right-hand side is in R-normal form, no word is the left-hand side of two different rewrite rules, and no left-hand side contains another left-hand side as a proper subword. (One can convert a convergent rewriting system R for G into its equivalent reduced convergent rewriting system R' for G such that $\overset{*}{\leftrightarrow}_R = \overset{*}{\leftrightarrow}_{R'}$ [6]). A rewriting system R for G (on S^*) is said to be *finite* if both S and R are finite sets.

Given a convergent rewriting system R for G, each of the group elements in G has its unique R-normal form. A convergent rewriting system for a finite group G provides a solution to the word problem for G, since each of the group elements can be represented by its unique R-normal form. A convergent rewriting system R for G is called a *complete presentation* for G.

Let $\prec$ be a shortlex order on S^*. A word $w \in S^*$ is the *lexicographically minimal reduced word* (or *the shortlex normal word*) for $g \in G$ if it is the minimal word w.r.t. $\prec$ on S^* that represents g. If a word $w \in S^*$ is the lexicographically minimal reduced word for some $g \in G$, then w is said to be in *shortlex normal form* (w.r.t. $\prec$) in G.

For each pair of not necessarily distinct rewriting rules from a rewriting system R for G on S^*, say (s_1, t_1) and (s_2, t_2), let the set of *critical pairs* corresponding to this pair be $\{ < xt_1, t_2 y > \mid \exists x, y \in S^*, xs_1 = s_2 y \text{ and } |x| < |s_2| \} \cup \{ < t_1, xt_2 y > \mid \exists x, y \in S^* \text{ with } s_1 = xs_2 y \}$. A critical pair $< z_1, z_2 >$ is said to be *resolved* in R if there is a word w such that $z_1 \overset{*}{\to} w$ and $z_2 \overset{*}{\to} w$. (Note that if a critical pair xt_1 and $t_2 y$ has two different normal forms w.r.t. R, then R is not confluent. A similar argument can be made for a critical pair t_1 and $xt_2 y$, which does not occur if R is reduced.)

The Knuth-Bendix (completion) procedure using the shortlex ordering creates a convergent rewriting system R for a finite group $G = < X \mid \mathcal{R} >$ with its monoid presentation $\text{Mon} < X_\alpha \mid I_X \cup \mathcal{R} >$. It relies on the fact that a Noetherian rewriting system is convergent if and only if every critical pair is resolved. It takes the set $I_X \cup \mathcal{R}$ of initial equations along with a shortlex order $\prec$ over X_α, and orients equations into rewrite rules w.r.t. $\prec$. Therefore, the initial set R of rewrite rules consists of the set of $s \to t$ for each equation $s = t$ in $I_X \cup \mathcal{R}$ with $t \prec s$. The Knuth-Bendix procedure then checks unresolved critical pairs in the rewriting system. If there is an unresolved critical pair $< z_1, z_2 >$, then rewrite z_1 and z_2 to words x and y that are in their normal forms, respectively, in the rewriting system. Then it adds to the rewriting system the

rule $x \to y$ if $y \prec x$ or $y \to x$ if $x \prec y$. This procedure continues until there are no more unresolved critical pairs in the rewriting system. The details of the Knuth-Bendix procedure with simplification and deletion are discussed in [34].

The Knuth-Bendix procedure necessarily terminates (in principle) for finite groups (with presentations) using the (total) shortlex ordering so that every finite group with its presentation admits a finite convergent rewriting system (see Corollary 12.21 in [26]).

A *Thue system*[2] is a rewriting system $T \subseteq S^* \times S^*$ such that the following conditions hold:
i) If $l \to r \in T$, then $|l| \geq |r|$. (By $|x|$ we mean the length of a string $x \in S^*$.)
ii) If $l \to r \in T$ with $|l| = |r|$, then $r \to l \in T$ too.

The *Thue resolution* of a rewriting system R is to symmetrize R by adding every rule $r \to l$ whenever $l \to r \in R$ and then to remove all the length increasing rules. A confluent Thue system is called *preperfect*.

Let G be a finite group and S be a generating set for G which is closed under inverses. The *Cayley graph* $\mathrm{Cay}(G, S)$ is a simple connected graph, where the vertices are the elements of G, and the edges are all ordered pairs (g, gs) for $g \in G, s \in S$. The resulting graph can be viewed as being undirected, meaning that the edges are directed in both ways (see [3, 32]).

The *bubble-sort Cayley graph* BS_n is the Cayley graph of the symmetric group $\mathfrak{S}_n$ generated by the adjacent transpositions $s_i = (i\ i{+}1)$, $1 \leq i < n$.

An *automorphism* of a simple connected graph $\Gamma = (V, E)$ is a permutation p of V such that (u, v) is an edge of Γ if and only if $(p(u), p(v))$ is an edge of Γ. The automorphisms of Γ form a subgroup of the group of all permutations of V and is denoted by $\mathrm{Aut}(\Gamma)$.

The *distance* from vertex u to vertex v in $\Gamma = (V, E)$, denoted by $d(u, v)$, is the length of a shortest path from u to v in $\Gamma = (V, E)$.

A simple connected graph $\Gamma = (V, E)$ is said to be *vertex transitive* (or *vertex symmetric*) if given any pair of vertices u and v, there exists an automorphism $\alpha \in \mathrm{Aut}(\Gamma)$ such that $v = \alpha(u)$, i.e., $\mathrm{Aut}(\Gamma)$ acts transitively on V.

A path p from vertex v_1 to vertex v_2 in $\mathrm{Cay}(G, S)$ can be written as a sequence of generators $g_1, \ldots, g_k$, $g_i \in S$, which can also be written as the word $p = g_1 \cdots g_k$. (For $k = 0$, p is interpreted as the empty path.)

In the remainder of this paper, $\mathrm{Cay}(G, S)$ denotes the Cayley graph of a finite group G with its generating set S which is closed under inverses.

3 Finding the Lexicographically Minimal Shortest Path Between Any Two Vertices in Cay(G, S)

The problem of finding a shortest path between two vertices in $\mathrm{Cay}(G, S)$ has been studied extensively over several decades (see, in particular, [3, 25, 27, 28].)

[2] There are slightly different definitions for Thue systems. We adopt the definition of the Thue system used in [19].

Cayley graphs are vertex-transitive graphs [4, 23, 28], which means that any Cayley graph looks the same regardless of what vertex it is viewed. Therefore, the same routing scheme can be adopted at each vertex of a Cayley graph.

Lemma 3.1 [3] *The problem of finding a shortest path from vertex v_1 to vertex v_2 in $Cay(G, S)$ is reduced to the problem of finding a shortest path from vertex 1 (i.e., the identity element of G) to vertex $v_1^{-1}v_2$, which in turn is equivalent to the problem of finding a minimum-length word (a reduced word) of $v_1^{-1}v_2$ over S in G.*

Since there are often multiple reduced words of $g \in G$ when written as a product of generators in S, we may have multiple shortest paths from one vertex to another in $\mathrm{Cay}(G, S)$. For example, there are two shortest paths s_1s_3 and s_3s_1 from vertex 1 to vertex $2143 \in \mathfrak{S}_4$ in the bubble-sort Cayley graph BS_4, where $s_i = (i\ i{+}1)$, $1 \leq i < 4$. When we consider a shortest path for $\mathrm{Cay}(G, S)$, we may need to choose a particular shortest path from vertex v_1 to vertex v_2 in $\mathrm{Cay}(G, S)$. We choose the lexicographically minimal shortest path among all shortest paths from vertex v_1 to vertex v_2 in $\mathrm{Cay}(G, S)$.

Definition 3.2 A path p from vertex v_1 to vertex v_2 in $\mathrm{Cay}(G, S)$ is a *shortlex normal path* if p is in shortlex normal form (w.r.t. a shortlex order $\prec$ on S^*) in G.

Note that every shortlex normal path is a shortest path, but the converse is not necessarily true. For example, with respect to the shortlex ordering induced by the lexicographic ordering $s_1 \prec s_2 \prec s_3$, we see that s_1s_3 is a shortlex normal path in BS_4, while s_3s_1 is a shortest path but it is not a shortlex normal path in BS_4.

Now, consider $\mathrm{Dih}_3 = < a, b \mid a^3 = 1, b^2 = 1, (ab)^2 = 1 >$ [26], which is a group presentation of the dihedral group of order 6. Its corresponding monoid presentation is $\mathrm{Mon} < a, a^{-1}, b, b^{-1} \mid aa^{-1} = 1, a^{-1}a = 1, bb^{-1} = 1, b^{-1}b = 1, a^3 = 1, b^2 = 1, (ab)^2 = 1 >$. Then we have an initial rewriting system $R' = \{aa^{-1} \to 1, a^{-1}a \to 1, bb^{-1} \to 1, b^{-1}b \to 1, a^2 \to a^{-1}, b^2 \to 1, b^{-1}a^{-1} \to ab\}$ for the above monoid presentation compatible with the shortlex ordering induced by the lexicographic ordering $a \prec a^{-1} \prec b \prec b^{-1}$. Throughout this paper, we assume that the length of the left-hand side is greater than the length of the right-hand side of each rewrite rule by at most 1 except the cases for $xy \to 1$, where x is the inverse of y and vice versa. This can be done by inverting the last element of the left-hand side and then transferring it to the end of the right-hand side of each rewrite rule whenever necessary. (See [34] for details on the "balance" heuristic to improve the performance of string rewriting systems). For example, $a^3 \to 1$ is converted to $a^2 \to a^{-1}$ as shown above. This is also intended not to miss any length-preserving rule for the Thue resolution of a reduced convergent rewriting system discussed in Sect. 4.

In order to find a reduced convergent rewriting system for R', we need to find all critical pairs in R' and orient them using the shortlex ordering. (One such critical pair is $< (a^{-1})^2, a >$, which is produced by overlapping two rules $a^2 \to a^{-1}$ and $a^{-1}a \to 1$ with the overlap a, which is resolved by orienting it into $(a^{-1})^2 \to a$.) Once we have a reduced convergent rewriting system for Dih_3, which is $R = \{aa^{-1} \to 1, a^{-1}a \to 1, a^2 \to a^{-1}, (a^{-1})^2 \to a, b^2 \to 1, b^{-1} \to b, ba \to a^{-1}b, ba^{-1} \to ab\}$ [26], we can find the unique shortlex normal form of each element $g \in \mathrm{Dih}_3$. It means that we can find the shortlex normal path from vertex v_1 to vertex

1: $aa \longrightarrow 1$
2: $bb \longrightarrow 1$
3: $cc \longrightarrow 1$
4: $bab \longrightarrow aba$
5: $cbc \longrightarrow bcb$
6: $ca \longrightarrow ac$
7: $cbac \longrightarrow bcba$

Fig. 1 The rewrite rules of a reduced convergent rewriting system R for the symmetric group $\mathfrak{S}_4$ generated by adjacent transpositions $a = (1\,2)$, $b = (2\,3)$, and $c = (3\,4)$ with the lexicographic ordering $a \prec b \prec c$

v_2 in Cay(Dih$_3$, $\{a, a^{-1}, b, b^{-1}\}$). For example, finding the shortlex normal path from vertex ab^{-1} to vertex $baba$ in Cay(Dih$_3$, $\{a, a^{-1}, b, b^{-1}\}$) is reduced to finding the shortlex normal word of $ba^{-1}baba$ in Dih$_3$ w.r.t. the shortlex ordering induced by the lexicographic ordering $a \prec a^{-1} \prec b \prec b^{-1}$. We have the following reduction steps for $ba^{-1}baba \in$ Dih$_3$ using the rewrite rules of R:

$$\underline{ba^{-1}}baba \to a\underline{b^2}aba \to \underline{a^2}ba \to a^{-1}\underline{ba} \to \underline{(a^{-1})^2}b \to ab,$$

which shows that the shortlex normal path from vertex ab^{-1} to vertex $baba$ in Cay(Dih$_3$, $\{a, a^{-1}, b, b^{-1}\}$) is ab.

Since a path p from vertex v_1 to vertex v_2 in Cay(G, S) is written as a product of generators $v_1^{-1}v_2 = g_1 \cdots g_k$, $g_i \in S$ by Lemma 3.1, a reduced convergent rewriting system R for G compatible with a shortlex order $\prec$ on S^* allows one to find the R-normal form of $v_1^{-1}v_2$. Let p' be the R-normal form of $v_1^{-1}v_2$. It is in shortlex normal form w.r.t. $\prec$ in G because R is compatible with $\prec$, and thus it is the shortlex normal path from v_1 to v_2 in Cay(G, S). Therefore, we have the following proposition.

Proposition 3.3 *Let v_1 and v_2 be two vertices in Cay(G, S). Then, we can find the shortlex normal path from v_1 to v_2 using a reduced convergent rewriting system R for G compatible with a shortlex order $\prec$ on S^*.*

4 Finding *k* Shortest Paths Between Any Two Vertices in Cay(*G*, *S*)

In this section, we show how to enumerate shortest paths between two vertices in Cay(G, S) starting with the shortlex normal path between two vertices in Cay(G, S) (see Proposition 3.3).

First, consider a group presentation of the symmetric group $\mathfrak{S}_4$ given by $\mathfrak{S}_4 = < a, b, c \,|\, aa = bb = cc = 1, aba = bab, bcb = cbc, ac = ca >$ as an example, where $a = (1\,2)$, $b = (2\,3)$, and $c = (3\,4)$ are adjacent transpositions. Here, the associated Cayley graph is the bubble-sort Cayley graph BS_4 with its generating set $S = \{a, b, c\}$. (Since $a = a^{-1}$, $b = b^{-1}$, and $c = c^{-1}$, S is closed under inverses.) We see that an initial rewriting system for $\mathfrak{S}_4$ w.r.t. the shortlex ordering induced by $a \prec b \prec c$ is $R' = \{aa \to 1, bb \to 1, cc \to 1, bab \to aba, cbc \to bcb, ca \to ac\}$.

Figure 1 shows the rewrite rules of a reduced convergent rewriting system R for $\mathfrak{S}_4$ compatible with the shortlex ordering induced by $a \prec b \prec c$ using R', obtained by our Geodesics tool (see Sect. 5 for details). Note that the rewriting rule shown in line

1: $aa \longrightarrow 1$
2: $bb \longrightarrow 1$
3: $cc \longrightarrow 1$
4: $bab \longrightarrow aba$
5: $aba \longrightarrow bab$
6: $cbc \longrightarrow bcb$
7: $bcb \longrightarrow cbc$
8: $ca \longrightarrow ac$
9: $ac \longrightarrow ca$
10: $cbac \longrightarrow bcba$
11: $bcba \longrightarrow cbac$

Fig. 2 The rewrite rules of the preperfect rewriting system T obtained from the Thue resolution of R in Fig. 1

7 of Fig. 1 has been added after a Knuth-Bendix procedure by resolving the critical pair produced by overlapping two rules $cbc \to bcb$ and $ca \to ac$ with the overlap c in R'.

Lemma 4.1 *The rewriting system obtained from the Thue resolution of a convergent rewriting system R for G (on S^*) compatible with a shortlex order $\prec$ on S^* is preperfect.*

Proof We see that R for G (on S^*) has no length-increasing rules because R is compatible with $\prec$. Since R is confluent and has no length-increasing rules, the Thue resolution of R is preperfect (see Lemma 4.5 in [19]). □

Figure 2 shows the rewrite rules of the preperfect rewriting system T obtained from the Thue resolution of R. It is easy to see that this rewriting system is not terminating and is not compatible with the shortlex ordering over $S = \{a, b, c\}$. However, we may use this rewriting system for enumerating reduced words over S in G. We first separate the length reducing rules (lines 1−3 in Fig. 2) and the length-preserving rules of T (lines 4−11 in Fig. 2).

Now, in order to find shortest paths from vertex u to vertex v in BS_4, we simply apply the possible length-preserving rules of T to the shortlex normal path for $u^{-1}v$. For example, we describe how to find all shortest paths from vertex ba to vertex $acba$ in BS_4. First, we see that the shortlex normal path from vertex ba to vertex $acba$ in BS_4 is simply $abacba$. (Here, $(ba)^{-1}acba = abacba$ and $abacba$ is in R-normal form (see Fig. 1).)

We use our level-search tree to enumerate alternative shortest paths for $abacba$, in which $abacba$ is the root (level 0) node of the level-search tree (see Fig. 3). Each node at level 1 of the level-search tree is found by using a single length-preserving rule application of T to the level 0 node (i.e., $abacba$) in the level-search tree. Therefore, $babcba$ and $abcaba$ are the nodes at level 1 of the level-search tree. Each node at level 2 of the level-search tree is found by using a single length-preserving rule application of T to each node at level 1 of the level-search tree, and so on. The visited nodes are maintained so that the level-search procedure does not go back to a previously visited node in the level-search tree. This process continues until no more new node can be discovered. Figure 3 shows all shortest paths from vertex ba to vertex $acba$ in BS_4, starting with the shortlex normal path $abacba$. Table 1 illustrates how each level

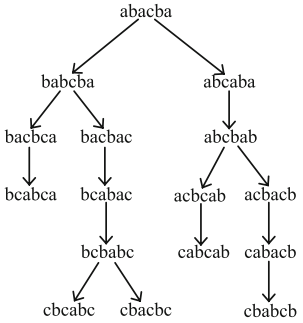

Fig. 3 The tree of shortest paths from vertex *ba* to vertex *acba* in the bubble sort graph BS_4 constructed by using the length-preserving rules in Fig. 2 and the level-search procedure. The root node is the shortlex normal path from vertex *ba* to vertex *acba* in BS_4

Table 1 Shortest paths and their applied rules for each level of the level-search tree shown in Fig. 3

Level	Shortest Path	Parent	Applied Rule
0	*abacba*		
1	*babcba*	*abacba*	$aba \longrightarrow bab$
	abcaba	*abacba*	$ac \longrightarrow ca$
2	*bacbca*	*babcba*	$bcb \longrightarrow cbc$
	bacbac	*babcba*	$bcba \longrightarrow cbac$
	abcbab	*abcaba*	$aba \longrightarrow bab$
3	*bcabca*	*bacbca*	$ac \longrightarrow ca$
	bcabac	*bacbac*	$ac \longrightarrow ca$
	acbcab	*abcbab*	$bcb \longrightarrow cbc$
	acbacb	*abcbab*	$bcba \longrightarrow cbac$
4	*bcbabc*	*bcabac*	$aba \longrightarrow bab$
	cabcab	*acbcab*	$ac \longrightarrow ca$
	cabacb	*acbacb*	$ac \longrightarrow ca$
5	*cbcabc*	*bcbabc*	$bcb \longrightarrow cbc$
	cbacbc	*bcbabc*	$bcba \longrightarrow cbac$
	cbabcb	*cabacb*	$aba \longrightarrow bab$

of shortest path(s) is obtained by using the level-search tree in Fig. 3 and the length-preserving rules shown in Fig. 2. For example, the shortest path *babcba* at level 1 in Table 1 is obtained by applying the length-preserving rule $aba \rightarrow bab$ in Fig. 2 to its parent node (parent shortest path) *abacba*.

Lemma 4.2 *Let $T \subset S^* \times S^*$ be a rewriting system obtained from the Thue resolution of a (finite) reduced convergent rewriting system $R \subset S^* \times S^*$ for G compatible with a shortlex order $\prec$ on S^*. For any finite word $w \in S^*$ in G and its shortlex normal word w_1 (w.r.t. $\prec$) in G, every reduced word (minimum-length word) w_k of w over S in G can be enumerated by using only the length-preserving part T_p of T such that*

$$w_1 \xrightarrow[Tp]{*} w_k.$$

Proof It is easy to see that each w_j satisfying $w_1 \xrightarrow[Tp]{*} w_j$ is a reduced word of w over S in G. As w_1 and T_p are both finite, we see that the total number of such w_j is finite. Now, we show that every reduced word w_k of w over S in G can be obtained by means of $w_1 \xrightarrow[Tp]{*} w_k$ only. For a contradiction, assume that there exists some reduced word w_r of w over S in G such that $w_1 \not\xrightarrow[Tp]{*} w_r$. As w_r is a reduced word of w over S in G, we have $w_1 \xleftarrow[T]{*} w \xrightarrow[T]{*} w_r$ with $|w_1| = |w_r|$. Since R is compatible with $\prec$, R does not have any length-increasing rule, and neither does T. By Lemma 4.1, T is preperfect, and thus confluent by the definition of a preperfect rewriting system. Therefore, we have $w_1 \xrightarrow[Tp]{*} w' \xleftarrow[Tp]{*} w_r$ for some reduced word w' of w over S in G. Since the rules of T_p are symmetric, we also have $w' \xrightarrow[Tp]{*} w_r$. It follows that $w_1 \xrightarrow[Tp]{*} w_r$, which is the required contradiction. □

Definition 4.3 The minimum number of rule applications of T_p required from w_1 to reach a reduced word w_k of w over S in G in Lemma 4.2 is the *depth of* w_k.

Definition 4.4 Let N be the total number of reduced words of w over S in G and let k be a positive integer such that $1 \leq k \leq N$. Then k reduced words of w over S in G are said to be *saturated* if the k reduced words contain every reduced word of w over S in G belonging to each depth i, $0 \leq i \leq M - 1$, where M is the maximum depth of the k reduced words of w over S in G if $k > 1$ and 1 otherwise.

For example, consider the shortlex normal word $abacba$ over $S = \{a, b, c\}$ in $\mathfrak{S}_4$ (see Fig. 3). Its four reduced words $abacba$, $abcaba$, $babcba$, and $abcbab$ over S in $\mathfrak{S}_4$ shown in Fig. 3 are saturated, while its four reduced words $abacba$, $abcaba$, $babcba$, and $acbcab$ over S in $\mathfrak{S}_4$ in Fig. 3 are not saturated. Now, the following corollary is immediate from Lemma 4.2 and Definition 4.4.

Corollary 4.5 *Let N be the total number of reduced words of w over S in G and w_1 be the shortlex normal word of w (w.r.t. $\prec$) in G. For enumerating k reduced words of w over S in G, we can always choose k saturated reduced words of w and enumerate them by means of $w_1 \xrightarrow[Tp]{*} w_i$, where $1 \leq i \leq k \leq N$.*

Now, we provide our algorithm for enumerating k shortest paths (if they exist) from vertex u to vertex v in $\mathrm{Cay}(G, S)$ based on Corollary 4.5. In the following, T_p is the length-preserving part of the preperfect rewriting system T obtained from the Thue resolution of a (finite) reduced convergent rewriting system R for G, where R is compatible with the shortlex ordering induced by a (total) lexicographic order $\prec$ on S.

Algorithm 4.6 ENUMERATION OF k SHORTEST PATHS (k, q, T_p) Input: A positive integer $k > 0$, the shortlex normal path q from vertex u to vertex v in $\mathrm{Cay}(G, S)$, and T_p. Output: Enumeration of k shortest paths found from vertex u to vertex v in $\mathrm{Cay}(G, S)$ if they exist. Otherwise, enumerate all shortest paths found from vertex u to vertex v in $\mathrm{Cay}(G, S)$.

- Set q as the root (level 0) node for the level-search tree of shortest paths from vertex u to vertex v in $\mathrm{Cay}(G, S)$. Set level 0 as the current level of the level-search tree.
- Do the following until the level-search tree has k nodes: For each node v at the current level of the level-search tree, find and add only a new node for the next level of the level-search tree by using a single rule application of T_p to v. This is repeated for v until the search of the left-hand sides in T_p is complete.
  - If no more length-preserving rule can be applied to the current level node(s), set the next level as the current level and continue the loop above. When setting the next level as the current level, if there is no new node for the next level (i.e., no new node can be discovered by using a single rule application of T_p to each node at the current level), then stop the loop.
- Enumerate the k shortest paths found from vertex u to vertex v in $\mathrm{Cay}(G, S)$ in the above level-search tree if they exist. Otherwise, enumerate all shortest paths found from vertex u to vertex v in $\mathrm{Cay}(G, S)$ in the above level-search tree.[3]

Theorem 4.7 *(a) Let $k > 0$ be a positive integer. Algorithm 4.6 correctly enumerates k shortest paths if they exist from vertex u to vertex v in $Cay(G, S)$.*
(b) Algorithm 4.6 runs in polynomial time in the size of k, $|q|$, and $||T_p||$ if k shortest paths exist from vertex u to vertex v in $Cay(G, S)$, where $|q|$ denotes the length of q and $||T_p||$ denotes $\sum_{(l,r)\in T_p}(|l| + |r|)$.

Proof Assume that k shortest paths exist from vertex u to vertex v in $\mathrm{Cay}(G, S)$. For the proof of part (a), we proceed by induction on k, the result being clear if $k = 1$ since it is the shortlex normal path q from vertex u to vertex v in $\mathrm{Cay}(G, S)$. So assume that $k > 1$ and that $q = p_1, p_2, \ldots, p_{k-1}$ are $k - 1$ distinct shortest paths from vertex u to vertex v added to the level-search tree sequentially by Algorithm 4.6. Since each edge in the level-search tree is associated with a rule of T_p, we have $q \xrightarrow[T_p]{*} p_i$, $1 \le i \le k-1$. As each shortest path from vertex u to vertex v in $\mathrm{Cay}(G, S)$ corresponds to each reduced word of $u^{-1}v$ over S in G and vice versa, it remains to show that a shortest path p_k, which is distinct from $q = p_1, p_2, \ldots, p_{k-1}$, is found by Algorithm 4.6 and is added to the level-search tree, i.e., $q \xrightarrow[T_p]{*} p_k$. By Lemma 4.2, this will prove the part (a) of the theorem.

We see that the level of each shortest path p_i, $1 \le i \le k-1$, in the level-search tree corresponds to the depth of p_i and that $p_1, \ldots, p_{k-1}$ are saturated. Now, if p_{k-1} is at the level l of the level-search tree, we have the following three different cases for a shortest path p_k by Corollary 4.5. Otherwise, it contradicts the assumption that k shortest paths exist from vertex u to vertex v in $\mathrm{Cay}(G, S)$. We denote the level i of the level-search tree by $\mathrm{LEVEL}(i)$.

Case 1: $p_s \xrightarrow[T_p]{} p_k$, where $p_s \in \mathrm{LEVEL}(l-1)$ such that $p_s \xrightarrow[T_p]{} p_{k-1}$.

Case 2: $p_t \xrightarrow[T_p]{} p_k$ and $p_u \xrightarrow[T_p]{} p_{k-1}$, where $p_t, p_u \in \mathrm{LEVEL}(l-1)$ and $t \neq u$.

[3] This is the case for example if k is given as a positive number n with $n > 0$, but there are only m shortest paths between two vertices u and v, where $m < n$.

Case 3: $p_r \xrightarrow[Tp]{} p_k$, where $p_r \in \text{LEVEL}(l)$.

By means of one of the above three cases, the level-search procedure in Algorithm 4.6 finds a shortest path p_k and adds it to the level-search tree, i.e., $q \xrightarrow[Tp]{*} p_k$.

For part (b) of the theorem, it takes polynomial time in the size of $|q|$ and $||T_p||$ for finding whether a shortest path p_i with $|p_i| = |q|$ contains the left-hand side of a rule of T_p, and replacing it with its right-hand side if a match occurs (see string matching algorithm in [16]). Note that for each node (i.e., shortest path) added to the level-search tree, Algorithm 4.6 searches the left-hand side of each rule of T_p sequentially. If a match occurs in the node, the left-hand side of the rule in the node is replaced by its right-hand side. If this rewriting step ends up with a previously found node in the level-search tree, which can be determined in polynomial time in the size of k and $|q|$, the algorithm restores the node and searches the left-hand side of the next rule of T_p. Otherwise, if this rewriting step finds a new node, the algorithm adds the new node to the level-search tree, restores the node, and searches the left-hand side of the next rule of T_p in order to find the next possible match in the node. This process continues until it completes the search of the left-hand sides in T_p, which takes in polynomial time in the size of k, $|q|$, and $||T_p||$. We may infer that finding new nodes one by one until the level-search tree has total k nodes also takes in polynomial time in the size of k, $|q|$, and $||T_p||$ using the level-search procedure and the construction of the level-search tree. Thus, the time complexity of Algorithm 4.6 runs in polynomial time in the size of k, $|q|$, and $||T_p||$. □

5 Discussion, Related Work, and Implementation

Given a graph $\Gamma = (V, E)$, the problem of enumerating shortest paths between two vertices in Γ is a well-known problem in graph theory, in which the solution of the problem has practical applications in interconnection networks in terms of routing flexibility, connectivity, and fault tolerance [14, 15, 33]. Although Cayley graphs are widely used as a mathematical framework for the design of symmetric interconnection networks [3, 25, 28, 32], it is surprising that few researches have been done to enumerate shortest paths between two vertices in Cayley graphs of finite groups in general.

In [20] Fortin et al. discussed a shortest path routing in $\text{Cay}(G, S)$ using rewriting techniques. Their approach is quite limited and is not applicable to enumerate shortest paths between any two vertices in $\text{Cay}(G, S)$.

The problem of counting/enumerating shortest paths (with or without constraints) between two vertices has been studied for certain types of graphs [8, 14, 15, 29]. Narraway [29] showed how to find alternative shortest paths in n-cubes. In [14] Cheng et al. discussed shortest paths between two vertices in arrangement graphs using *reduced decompositions* [30] of permutations in terms of *arrangement transpositions* [14]. See [15] also for the case of (n, k)-star graphs. However, their methods are not applicable to enumerate shortest paths between any two vertices in $\text{Cay}(G, S)$ in general.

In [2], Aguirre-Guerrero et al. discussed path computation including the shortest paths computation in Cayley graphs using an automaton called *Diff*, which has also the scalability issue for $\mathrm{Cay}(G, S)$ in general (cf. Table 17 in [1]).

Recall that various sets of transpositions generate the symmetric group $\mathfrak{S}_n$. Pak [30] considered reduced decompositions of permutations in terms of star transpositions, while Stanley [35] discussed reduced decompositions of permutations in terms of adjacent transpositions. If S is a set of transpositions generating $\mathfrak{S}_n$, it can be represented by the *transposition graph* $T(S)$ [25, 28]. (The transposition graph $T(S)$ is an undirected graph with vertex set $\{1, \ldots, n\}$ and with vertices i and j being adjacent in $T(S)$ if and only if $(i\ j) \in S$.) Note that reduced decompositions of a permutation $p \in \mathfrak{S}_n$ in terms of transpositions represented by $T(S)$ can be obtained by finding shortest paths from vertex 1 to vertex p in $\mathrm{Cay}(\mathfrak{S}_n, S)$. Furthermore, the group presentation of $\mathfrak{S}_n$ w.r.t. S can be constructed explicitly from $T(S)$ (see Theorem 1 in [18]). This means that our rewriting approach allows one to enumerate reduced decompositions of permutations in terms of transpositions represented by a wide variety of transposition graphs.

Now, we briefly describe our implementation, called Geodesics.[4] The purpose of our implementation is to provide a tool for finding the shortlex normal path and enumerating k shortest paths (if they exist) between two vertices in $\mathrm{Cay}(G, S)$. We referred to the book by Sims [34] for a Knuth-Bendix procedure and its related heuristics, and referred to the paper by Diekert et al. [19] for the Thue resolution. Our implementation had been developed using the standard GNU C++ language [21]. In the "data" directory of our source codes we provide the sample input and output files for different Cayley graphs with their finite group presentations. The input of our implementation consists of the source and target vertex in $\mathrm{Cay}(G, S)$, a presentation of G w.r.t. a lexicographically ordered generating set S, the number k for finding k shortest paths (if they exist) from the source vertex to the target vertex in $\mathrm{Cay}(G, S)$, etc. If the number k is set to 0, the program enumerates every shortest path found from the source vertex to the target vertex in $\mathrm{Cay}(G, S)$, provided that it terminates within a given specified amount of time. Meanwhile, the output of our implementation yields a reduced convergent rewriting system for G along with its Thue resolution, the shortlex normal path and k shortest paths (if they exist) from the source vertex to the target vertex in $\mathrm{Cay}(G, S)$, etc. Now, we describe some example outputs obtained by Geodesics. In the following examples, since S is closed under inverses, the edges of $\mathrm{Cay}(G, S)$ are directed in both ways (i.e., undirected).

Example 5.1 The bubble-sort Cayley graph BS_5 of $\mathfrak{S}_5 = \langle a, b, c, d \mid aa = bb = cc = dd = 1, aba = bab, bcb = cbc, cdc = dcd, ac = ca, ad = da, bd = db \rangle$, where $a = (1\,2)$, $b = (2\,3)$, $c = (3\,4)$, and $d = (4\,5)$ with the lexicographic ordering $a \prec b \prec c \prec d$:

BS_5 has 120 vertices and 240 undirected edges. The output of Geodesics had 18 length-preserving rules in the preperfect rewriting system obtained from the Thue resolution of a reduced convergent rewriting system for the above presentation of $\mathfrak{S}_5$. It had 768 shortest paths from 1 to $5\,4\,3\,2\,1 \in \mathfrak{S}_5$, where $5\,4\,3\,2\,1 \in \mathfrak{S}_5$ is the longest

[4] Source codes and data for our implementation are freely available under the GPL license [22] at https://github.com/SortingOnGraphs/Geodesics.

element in $\mathfrak{S}_5$ [35]. (The permutation 5 4 3 2 1 $\in \mathfrak{S}_5$ can be converted into the word in the generators a, b, c, and d by using the bubble-sort algorithm. See [27] for details.) Note that the number of shortest paths from vertex 1 to vertex $n\,n\text{-}1\cdots 2\,1 \in \mathfrak{S}_n$ in BS_n coincides with the number of reduced decompositions of the longest element $n\,n\text{-}1\cdots 2\,1 \in \mathfrak{S}_n$ w.r.t. the generating set $\{(i\ i{+}1) \mid 1 \leq i < n\}$, which is given as follows [35]:

$$\frac{\binom{n}{2}!}{1^{n-1}3^{n-2}5^{n-3}\cdots(2n-3)^1}.$$

The above formula gives the value 16 for $n = 4$ and the value 768 for $n = 5$, respectively. Indeed, $abacba$ in Fig. 3 corresponds to the longest element 4 3 2 1 $\in \mathfrak{S}_4$.

Example 5.2 The star graph ST_5 of $\mathfrak{S}_5 = < a, b, c, d \mid aa = bb = cc = dd = 1, aba = bab, aca = cac, ada = dad, bcb = cbc, bdb = dbd, cdc = dcd, abcb = bcba, abdb = bdba, bcdc = cdcb, acdc = cdca >$ (see [18]), where $a = (1\,2), b = (1\,3), c = (1\,4)$, and $d = (1\,5)$ with the lexicographic ordering $a \prec b \prec c \prec d$:

The output of Geodesics had 7 length decreasing rules and 126 length-preserving rules in the preperfect rewriting system obtained from the Thue resolution of a reduced convergent rewriting system for the above presentation of $\mathfrak{S}_5$. It had 24 shortest paths from vertex 1 to vertex $abacdc$. Table 2 shows all the shortest paths from vertex 1 to vertex $abacdc$ in ST_5 obtained by Geodesics, which also shows each level of the corresponding level-search tree.

Example 5.3 The Cayley graph $\mathrm{Cay}(D_4, \{a, b, c, d\})$ of the Coxeter group $D_4 = < a, b, c, d \mid a^2 = b^2 = c^2 = d^2 = (ac)^3 = (bc)^3 = (cd)^3 = (ab)^2 = (ad)^2 = (bd)^2 = 1 >$ with the lexicographic ordering $a \prec b \prec c \prec d$ [7]:

As the order of Coxeter group $D_n, n \geq 4$, is $2^{n-1}n!$ [5, 7], $\mathrm{Cay}(D_4, \{a, b, c, d\})$ has 192 vertices and 384 undirected edges. A reduced convergent rewriting system R for D_4 obtained by Geodesics was as follows: $R = \{aa \to 1, bb \to 1, cc \to 1, dd \to 1, cac \to aca, cbc \to bcb, dcd \to cdc, ba \to ab, da \to ad, db \to bd, cabca \to bcabc, cabcb \to acabc, dcad \to cdca, dcbd \to cdcb, dcabd \to cdcab, dcabcdc \to cdcabcd\}$. It had 16 rewrite rules, which coincides with a reduced convergent rewriting system for D_4 shown in [7]. Since there are 12 length-preserving rules of the above rewriting system, the output of Geodesics had 24 length-preserving rules in the preperfect rewriting system obtained from the Thue resolution of R. It also had total 46 shortest paths from vertex $abac$ to vertex $adcabc$ in $\mathrm{Cay}(D_4, \{a, b, c, d\})$.

Example 5.4 The Cayley graph $\mathrm{Cay}(B_6, \{a, b, c, d, e, f\})$ of the Weyl group $B_6 = < a, b, c, d, e, f \mid a^2 = b^2 = c^2 = d^2 = e^2 = f^2 = (ab)^3 = (ac)^2 = (ad)^2 = (ae)^2 = (af)^2 = (bc)^3 = (bd)^2 = (be)^2 = (bf)^2 = (cd)^3 = (ce)^2 = (cf)^2 = (de)^3 = (df)^2 = (ef)^4 = 1 >$ [13] with the lexicographical ordering $a \prec b \prec c \prec d \prec e \prec f$:

As the order of Weyl group of type B_6 is 46,080 [13], $\mathrm{Cay}(B_6, \{a, b, c, d, e, f\})$ has 46,080 vertices and 138,240 undirected edges. The output of Geodesics had 132

Table 2 Enumeration of all the shortest paths from vertex 1 to vertex $abacdc$ in ST_5 obtained by Geodesics

Level	Shortest Path	Parent	Applied Rule
0	$abacdc$		
1	$babcdc$	$abacdc$	$aba \longrightarrow bab$
	$abadcd$	$abacdc$	$cdc \longrightarrow dcd$
	$cabadc$	$abacdc$	$abac \longrightarrow caba$
	$abcdca$	$abacdc$	$acdc \longrightarrow cdca$
2	$abdcda$	$abcdca$	$cdc \longrightarrow dcd$
	$acdcba$	$abcdca$	$bcdc \longrightarrow cdcb$
	$cbabdc$	$cabadc$	$aba \longrightarrow bab$
	$cdabac$	$cabadc$	$abad \longrightarrow daba$
	$babdcd$	$abadcd$	$aba \longrightarrow bab$
	$dabacd$	$abadcd$	$abad \longrightarrow daba$
	$bacdcb$	$babcdc$	$bcdc \longrightarrow cdcb$
3	$badcdb$	$bacdcb$	$cdc \longrightarrow dcd$
	$bcdcab$	$bacdcb$	$acdc \longrightarrow cdca$
	$dbabcd$	$dabacd$	$aba \longrightarrow bab$
	$dcabad$	$dabacd$	$abac \longrightarrow caba$
	$cdbabc$	$cdabac$	$aba \longrightarrow bab$
	$cdcaba$	$cdabac$	$abac \longrightarrow caba$
	$adcdba$	$acdcba$	$cdc \longrightarrow dcd$
4	$cdcbab$	$cdcaba$	$aba \longrightarrow bab$
	$dcdaba$	$cdcaba$	$cdc \longrightarrow dcd$
	$dcbabd$	$dcabad$	$aba \longrightarrow bab$
	$bdcdab$	$bcdcab$	$cdc \longrightarrow dcd$
5	$dcdbab$	$dcdaba$	$aba \longrightarrow bab$

length-preserving rules in the preperfect rewriting system obtained from the Thue resolution of a reduced convergent rewriting system for B_6. Note that the number of length-preserving rules here is significantly smaller than the order of B_6 (cf. Theorem 4.7 (b)). It also had total 48 shortest paths from vertex 1 to vertex $abcdefabc$ in $\mathrm{Cay}(B_6, \{a, b, c, d, e, f\})$.

Finally, we conclude this paper by discussing a future research direction. One possible future research direction is to apply the variants of our rewriting-based approach for finding k shortest paths in $\mathrm{Cay}(G, S)$ to certain classes of graphs other than $\mathrm{Cay}(G, S)$. In this case, one may need some labeling functions on edges and the associated rewrite rules on them for those graphs.

Funding Open access funding provided by University of Innsbruck and Medical University of Innsbruck. The author has no relevant funding or grants to disclose.

Declarations

Conflict of interest The author has no relevant financial or non-financial interests to disclose.

References

1. Aguirre-Guerrero, D.: Word-processing-based routing for Cayley graphs. Doctoral thesis, University of Girona, Girona, Catalonia, Spain (2019)
2. Aguirre-Guerrero, D., Ducoffe, G., Fàbrega, L., Vilà, P., Coudert, D.: Low time complexity algorithms for path computation in Cayley Graphs. Discrete Appl. Math. **259**, 218–225 (2019)
3. Akers, S.B., Krishnamurthy, B.: A group-theoretic model for symmetric interconnection networks. IEEE Trans. Comput. **38**(4), 555–566 (1989)
4. Biggs, N.: Algebraic Graph Theory, 2nd edn. Cambridge University Press, Cambridge (1993)
5. Björner, A., Brenti, F.: Combinatorics of Coxeter Groups. Springer, New York (2005)
6. Book, R.V., Otto, F.: String-Rewriting Systems. Springer, New York (1993)
7. Borges-Trenard, M.A., Pérez-Rosés, H.: Complete presentations of Coxeter groups. Appl. Math. E-Notes **4**, 1–6 (2004)
8. Botea, A., Mattetti, M., Kishimoto, A., Marinescu, R., Daly, E.: Counting Vertex-Disjoint Shortest Paths in Graphs. In: Proceedings of the Fourteenth International Symposium on Combinatorial Search, SOCS 2021, pp. 28–36 (2021)
9. Butler, G.: Fundamental Algorithms for Permutation Groups. Springer, New York (1991)
10. Camelo, M., Papadimitriou, D., Fàbrega, L., Vilà, P.: efficient routing in data center with underlying Cayley graph. In: Complex Networks V - Proceedings of the 5th Workshop on Complex Networks CompleNet 2014, Bologna, Italy, March 12–14, 2014, pp. 189–197 (2014)
11. Camelo, M., Vilà, P., Fàbrega, L., Papadimitriou, D.: Cayley-graph-based data centers and space requirements of a routing scheme using automata. In: Proceedings of the 2014 IEEE 34th International Conference on Distributed Computing Systems Workshops, ICDCSW '14, pp. 63–69 (2014)
12. Cannon, J.J.: Construction of defining relators for finite groups. Discrete Math. **5**(2), 105–129 (1973)
13. Cannon, J.J., Dimino, L.A., Havas, G., Watson, J.M.: Implementation and analysis of the Todd-Coxeter algorithm. Math. Comput. **27**(123), 463–490 (1973)
14. Cheng, E., Grossman, J.W., Qiu, K., Shen, Z.: The number of shortest paths in the arrangement graph. Inf. Sci. **240**, 191–204 (2013)
15. Cheng, E., Qiu, K., Shen, Z.: The number of shortest paths in the (n, k)-star graph. Discrete Math. Algor. Appl. **06**(04), 1450051 (2014)
16. Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C.: Introduction to Algorithms, 2nd edn. The MIT Press, Cambridge (2001)
17. Dekov, D.V.: Finite complete rewriting systems for groups. Commun. Algebra **25**(12), 4023–4028 (1997)
18. Delorme, C.: Presentations of groups generated by transpositions. Discrete Math. **236**(1), 59–64 (2001)
19. Diekert, V., Duncan, A.J., Myasnikov, A.G.: Geodesic rewriting systems and pregroups. In: Bogopolski, O., Bumagin, I., Kharlampovich, O., Ventura, E. (eds.) Combinatorial and Geometric Group Theory, Trends in Mathematics, pp. 55–91. Birkhäuser, Basel (2010)
20. Fortin, D., Kirchner, C., Strogova, P.: Routing in regular networks using rewriting. In: J. Slaney (ed.) Proceedings of the CADE International Workshop on Automated Reasoning in Algebra (ARIA), pp. 5–8 (1994)
21. Free Software Foundation: The GNU C++ Library (libstdc++ documentation). http://gcc.gnu.org/onlinedocs/libstdc++

22. Free Software Foundation: GNU General Public License, Version 3 (2007). http://www.fsf.org/copyleft/gpl.html
23. Godsil, C., Royle, G.: Algebraic Graph Theory. Springer, New York (2001)
24. Hermiller, S.M.: Rewriting systems for Coxeter groups. J. Pure Appl. Algebra **92**(2), 137–148 (1994)
25. Heydemann, M.: Cayley graphs and interconnection networks. In: Hahn, G., Sabidussi, G. (eds.) Graph Symmetry: Algebraic Methods and Applications, NATO ASI Series, vol. 497, pp. 167–224. Springer, New York (1997)
26. Holt, D.F., Eick, B., O'Brien, E.A.: Handbook of Computational Group Theory. CRC Press, Boca Raton (2005)
27. Kim, D.: Sorting on graphs by adjacent swaps using permutation groups. Comput. Sci. Rev. **22**, 89–105 (2016)
28. Lakshmivarahan, S., Jwo, J.S., Dhall, S.K.: Symmetry in interconnection networks based on Cayley graphs of permutation groups: a survey. Parallel Comput. **19**(4), 361–407 (1993)
29. Narraway, J.J.: Alternative shortest paths in n-cubes. Electron. Lett. **36**(23), 1916–1918 (2000)
30. Pak, I.: Reduced decompositions of permutations in terms of star transpositions, generalized Catalan numbers and k-ARY trees. Discrete Math. **204**, 329–335 (1999)
31. Rosen, K.H.: Discrete Mathematics and Its Applications, 3rd edn. McGraw-Hill, New York (1995)
32. Schibell, S.T., Stafford, R.M.: Processor interconnection networks from Cayley graphs. Discrete Appl. Math. **40**(3), 333–357 (1992)
33. Schwiebert, L.: There is no optimal routing policy for the torus. Inf. Process. Lett. **83**(6), 331–336 (2002)
34. Sims, C.C.: Computation with Finitely Presented Groups. Cambridge University Press, New York (1994)
35. Stanley, R.P.: On the number of reduced decompositions of elements of Coxeter groups. Eur. J. Combin. **5**(4), 359–372 (1984)
36. Strogova, P.: Finding a finite group presentation using rewriting. In: Bronstein, M., Weispfenning, V., Grabmeier, J. (eds.) Symbolic Rewriting Techniques, Progress in Computer Science and Applied Logic, vol. 15, pp. 267–276. Birkhäuser, Reinach (1998)