Contributions to Discrete Mathematics

Volume 7, Number 2, Pages 105–118
ISSN 1715-0868

ALGORITHMS FOR CLASSIFYING REGULAR POLYTOPES WITH A FIXED AUTOMORPHISM GROUP

DIMITRI LEEMANS AND MARK MIXER

Abstract. In this paper, various algorithms used in the classifications of regular polytopes for given groups are compared. First computational times and memory usages are analyzed for the original algorithm used in one of these classifications. Second, a possible algorithm for isomorphism testing among polytopes is suggested. Then, two improved algorithms are compared, and finally, results are given for a new classification of all regular polytopes for certain alternating groups and for the third Conway group, Co_3.

1. Introduction

In order to gain insight into the structure of a group, one natural method is to study some geometric and combinatorial objects on which the group acts. Abstract regular polytopes are highly symmetric combinatorial structures with distinctive geometric, algebraic, or topological properties, and thus work well for this purpose. Additionally, using the correspondence between abstract regular polytopes and string C-groups, finding abstract regular polytopes for a group gives a presentation of the group as generated by a set of involutions with many nice properties.

In 2006, D. Leemans and L. Vauthier published "An atlas of polytopes for almost simple groups" [12], classifying all regular polytopes for almost simple groups as large as the automorphism group of a simple group with 900,000 elements. Similarly in 2006, M. Hartley published "An atlas of small regular polytopes" [8], where he classified all regular polytopes for all groups of order at most 2000 (not including orders 1024 and 1536). More recently in [9], M. Hartley and A. Hulpke classified all polytopes for the sporadic groups as large as the Held group (of order 4,030,387,200).

These atlases, and other computational data alike, have lead to many conjectures (and eventually many recent theorems) in the field of abstract polytopes. For example, one can observe that, for small q, there are only 2 polytopes of rank 4 and no polytopes of rank greater than 4 for the groups

Received by the editors November 21, 2011, and in revised form September 25, 2012.
2000 *Mathematics Subject Classification.* 51M20, 20D08, 20F65, 52B15.
Key words and phrases. Abstract regular polytopes, third Conway group, alternating groups.

isomorphic to $PSL(2,q)$. In [10], starting from this observation, the complete classification was determined for polytopes of rank at least 4, with automorphism group isomorphic to a $PSL(2,q)$ group. Similar work was done in [11], for the groups of types $G \cong PGL(2,q)$, and in [2] for groups G, $PSL(3,q) \leq G \leq PGL(3,q)$.

Additionally, the computational data lead to conjectures about the symmetric groups which were proved in [4], and conjectures about the alternating groups proved in [6] and [5]. In these papers regarding the alternating groups, patterns in the data were not clear for groups of small degree, thus better computational data were needed to understand the structure in these cases. This provided a motivation to improve the algorithms of [12], and to make available better computational data for groups too large for previous techniques.

The improvements in this computational approach were already useful in proving theorems about the alternating groups, and should prove to be helpful for others studying other series of groups which grow quickly in size.

The main result of our paper is the verification of the classifications of Hartley and Hulpke using algorithms written independently in Magma [1] (rather than GAP [7]). Another result is the classification of all polytopes for Conway's third group Co_3 (of order 495,766,656,000) and for alternating groups of degree up to 14. Finally, we also give a possible approach for testing isomorphism on polytopes using CPR-graphs. This isomorphism-test, although not needed by the algorithms as they ensure that non-isomorphic polytopes are constructed, may well turn to be very useful if we want to check that two given polytopes are isomorphic.

In Section 2 we give an overview of abstract regular polytopes and string C-groups. Next in Section 3, we breakdown how much time and memory is used in the algorithms of [12]. In Section 4, we consider a possible approach for testing isomorphism of polytopes using CPR-graphs. In Section 5, we present the breadth-first and depth-first search algorithms described in [9], and compare their efficiency. Then in Sections 6 and 7, we summarize a new classification of the regular polytopes with automorphism group an alternating group A_n for $10 \leq n \leq 14$, and regular polytopes with automorphism group the sporadic simple group Co_3.

2. Regular polytopes and string C-groups

An *abstract d-polytope* $\mathcal{P}$ is a ranked partially ordered set of *faces* with the following four defining properties (see [13] for more details). First, $\mathcal{P}$ contains two improper faces, a least face F_{-1} of rank -1, and a greatest face F_d of rank d; in general, an element $F \in \mathcal{P}$ with $\mathrm{rank}(F) = i$ is called a *i-face*. Second, each *flag* (maximal totally ordered subset) of $\mathcal{P}$ contains $d+2$ faces (including the two improper faces). Third, $\mathcal{P}$ is strongly connected, in the sense defined below. Finally, $\mathcal{P}$ must have a homogeneity property;

whenever $F < G$ with $\text{rank}(F) = i-1$ and $\text{rank}(G) = i+1$, there are exactly two i-faces H with $F < H < G$; this is called the diamond property.

For any two faces F and G of $\mathcal{P}$ with $F \leq G$, we call $G/F := \{H \mid H \in \mathcal{P},\ F \leq H \leq G\}$ a *section* of $\mathcal{P}$; this is a polytope in its own right. If $\mathcal{P}$ is a partially ordered set satisfying the first two properties, then $\mathcal{P}$ is said to be *connected* if either $d \leq 1$ or $d \geq 2$, and for any two *proper* faces F and G of $\mathcal{P}$ (meaning any faces other than F_{-1} and F_d) there is a sequence of proper faces $F = H_0, H_1, \ldots, H_{k-1}, H_k = G$ such that H_i and H_{i-1} are comparable for $i = 1, \ldots, k$. We say that $\mathcal{P}$ is *strongly connected* if each section of $\mathcal{P}$ (including $\mathcal{P}$ itself) is connected.

Two flags of a d-polytope $\mathcal{P}$ are said to be *adjacent* if they differ by exactly one face. If Φ is a flag of $\mathcal{P}$, the diamond property tells us that for $i = 0, 1, \ldots, d-1$ there is exactly one flag that differs from Φ in its i-face. This flag is denoted Φ^i and is *i-adjacent* to Φ. Note that $(\Phi^i)^i = \Phi$ for each i, and $(\Phi^i)^j = (\Phi^j)^i$ if $|i-j| > 1$. A d-polytope ($d \geq 2$) is called *equivelar* if for each $i = 1, 2, ..., d-1$, there is an integer p_i so that any section G/F defined by an $(i-2)$-face F and an $(i+1)$-face G is a p_i-gon. If $\mathcal{P}$ is equivelar we say that it has (*Schläfli*) *type* $\{p_1, p_2, \ldots, p_{d-1}\}$.

The automorphism group of a d-polytope $\mathcal{P}$ is denoted by $\Gamma(\mathcal{P})$. A d-polytope $\mathcal{P}$ is called *regular* if its automorphism group $\Gamma(\mathcal{P})$ has exactly one orbit on the flags of $\mathcal{P}$, or equivalently, if for some flag $\Phi = \{F_{-1}, F_0, F_1, \ldots, F_d\}$ and each $i = 0, 1, \ldots, d-1$ there exists a (unique involutory) automorphism ρ_i of $\mathcal{P}$ such that $\rho_i(\Phi) = \Phi^i$. If $\mathcal{P}$ is regular, then in fact the latter property holds for any flag Φ, and so we are free to choose any fixed, or *base*, flag as a reference flag.

For a regular d-polytope $\mathcal{P}$ with base flag Φ, its group $\Gamma(\mathcal{P})$ is generated by the involutions $\rho_0, \ldots, \rho_{d-1}$ described above. These *distinguished generators* satisfy the Coxeter-type relations

$$(\rho_i\rho_j)^{p_{ij}} = \epsilon \quad (0 \leq i, j \leq d-1), \tag{1}$$

where $p_{ii} = 1$ for all i, $2 \leq p_{ji} = p_{ij} \leq \infty$ if $j = i-1$, and with the additional property that

$$p_{ij} = 2 \text{ for } |i-j| \geq 2. \tag{2}$$

Any group $\langle \rho_0, \ldots, \rho_{d-1} \rangle$ satisfying properties 1 and 2 is called a *String Group Generated by Involutions* or an *sggi*. Here the p_i's are given by the type $\{p_1, p_2, ..., p_{d-1}\}$ of $\mathcal{P}$. Moreover, $\Gamma(\mathcal{P})$ and its generators satisfy the following *intersection property*:

$$\langle \rho_i \mid i \in I \rangle \cap \langle \rho_i \mid i \in J \rangle = \langle \rho_i \mid i \in I \cap J \rangle \text{ for } I, J \subseteq \{0, \ldots, d-1\}. \tag{3}$$

Any group generated by involutions that has this intersection property is called a *C-group* (see [13, Ch. 2]). Thus the group $\Gamma(\mathcal{P})$ of a regular polytope $\mathcal{P}$ is always a C-group from what was mentioned above. However, not all C-groups are Coxeter groups as there can be other, independent relations amongst the generators. Note that property 2 implies that the underlying

Coxeter diagram for $\Gamma(\mathcal{P})$ is a string diagram; thus $\Gamma(\mathcal{P})$ is always what is called a *string C-group*.

Conversely, in [13], it is proved that a regular d-polytope can be constructed uniquely from a string C-group $\Gamma = \langle \rho_0, \ldots, \rho_{d-1} \rangle$. To see how this is accomplished we need further notation. Let $\Gamma_i := \langle \rho_j | j \neq i \rangle$, that is for each i, Γ_i is the group generated by all but the i^{th} generator. The faces of $\mathcal{P}$ and the partial order are defined in terms of cosets of these groups. First set $\Gamma_{-1} := \Gamma_d := \Gamma := \Gamma(\mathcal{P})$. Next, for all ranks j, take the set of j-faces of $\mathcal{P}$ to be the set of all right cosets $\Gamma_j \varphi$ in Γ, with $\varphi \in \Gamma$. Finally, define the partial order by $\Gamma_j \varphi \leq \Gamma_k \psi$ if and only if $-1 \leq j \leq k \leq d$ and $\Gamma_j \varphi \cap \Gamma_k \psi \neq \emptyset$. This construction identifies $\mathcal{P}$ as a particular kind of *thin diagram geometry* (see [3]).

3. First Results

In the following table we compare how much time and memory is required to find all regular polytopes for a given group G using the algorithms from [12] that classify all polytopes up to isomorphism and duality. The first column contains the name of the group considered, the second column gives its order and the third column gives the number of polytopes found, up to isomorphism and duality. The "Time 1" column gives the time taken by our Magma implementation of the algorithms to classify these polytopes. In "Time 2", we slightly modify those algorithms to remove the isomorphism/duality test. The last two columns give us the memory usage of each implementation. Thus, in "Time 2" some polytopes may be found multiple times and some isomorphism check still needs to be performed to finish the task. This gives us an approximation of how much time/memory was used to find the polytopes, and how much time/memory was used to test for isomorphisms and dualities.

Group	Order	Polytopes	Time 1	Time 2	Memory 1	Memory 2
$Alt(5)$	60	2	0.17s	0.18s	0.01MB	0.01MB
$Alt(5) \times C_2$	120	8	0.22s	0.29s	0.01MB	0.01MB
$P\Gamma L(2,9)$	1440	12	1.22s	1.86s	1.04MB	0.01MB
$Sym(7)$	5040	44	5.4s	5.95s	3.51MB	1.04MB
$PSL(2,25)$	7800	17	6.27s	0.39s	20.92MB	0.01MB
$P\Sigma U(3,3)$	12096	31	10.82s	4.62s	8.73MB	1.29MB
$PGL(2,27)$	19656	98	87.83s	4.38s	39.64MB	1.04MB
$Sz(8)$	29120	7	10.45s	0.44s	60.64MB	1.29MB
M_{12}	95040	37	224.59s	12.19s	126.2MB	2.32MB
J_1	175560	150	550.85s	54.36s	120.17MB	3.95MB
$Alt(9)$	181440	47	654.65s	37.12s	245.64MB	2.32MB
$Sym(9)$	362880	182	8152.22s	2193.75s	261.72MB	10.26MB

As a technical note, the memory data in this table have been adjusted by subtracting the 9.03MB of memory required to load the current version of

Magma, and the time data are the differences in CPU time from start to end of the calculations.

From this data we observe that, as the groups get larger, much of the time and memory for finding polytopes for a given group comes in the check for isomorphisms and dualities. This leads us to try and find a more efficient test to check whether two polytopes are equivalent or not.

4. CPR graphs and isomorphisms of polytopes

In this section we define CPR graphs for regular polytopes, introduced by Pellicer in [14]. A *CPR graph* of a regular d-polytope $\mathcal{P}$ is a permutation representation of $\Gamma(\mathcal{P}) = \langle \rho_0, \ldots, \rho_{d-1} \rangle$ represented on a graph as follows. Let ϕ be an embedding of $\Gamma(\mathcal{P})$ into the symmetric group S_n for some n. The CPR graph G of $\mathcal{P}$ determined by ϕ is the multigraph with n vertices, and with edge labels in the set $\{0, \ldots, d-1\}$, such that any two vertices v, w are joined by an edge of label j if and only if $(v)(\rho_j)\phi = w$. These representations are faithful since ϕ is an embedding.

The value of n for which we choose an embedding into S_n is not unique. We prove that one can test for isomorphism of two regular polytopes by testing for isomorphisms of their CPR graphs given an appropriate embedding. Whether a test finds only isomorphisms or also dualities, depends on the definition of an isomorphism between two edge labeled multigraphs. If we consider the edge labels with an order, and for example, only allow an edge of label 1 to be mapped to another edge of label 1, then we have a test for isomorphism. If we allow that the labels be permuted, then we have a test for isomorphism and duality. We use the first idea of isomorphism in our proof, and test for duality by testing for isomorphism with the order of the labels reversed.

Theorem 4.1. *Let G be any group, and let G and $Aut(G)$ be embedded as subgroups of $Sym(n)$ for some n. Then two regular polytopes $\mathcal{P}$ and $\mathcal{P}'$ with automorphism group G are isomorphic if and only if their CPR graphs (with the trivial embedding) are isomorphic as edge labeled graphs.*

Proof. Let $\mathcal{P} = \langle \rho_0, \ldots, \rho_{r-1} \rangle$, $\mathcal{P}' = \langle \rho'_0, \ldots, \rho'_{r-1} \rangle$ for the group G, and let X_1 be the CPR graph for $\mathcal{P}$ and X_2 be the CPR graph for $\mathcal{P}'$.

($\Rightarrow$) Let $\mathcal{P}$ and $\mathcal{P}'$ be isomorphic regular polytopes for the group G. Then there exists an $\alpha \in Aut(G)$ such that $\rho_i \alpha = \rho'_i$ for all $i \in \{0, \ldots, r-1\}$. Since G and $Aut(G)$ are considered as subgroups of $Sym(n)$ we know that ρ_i and ρ'_i have the same cycle type, as the action of $Aut(G)$ on G is by conjugation. Also, since ρ_i and ρ'_i are involutions, we can see α as acting on the transpositions in the cycle decomposition of ρ_i. Each such transposition gives an edge in X_1, and thus α sends edges of label i in X_1 to edges of label i in X_2. Therefore X_1 and X_2 are isomorphic as edge labeled graphs.

($\Leftarrow$) Let $X_1 \cong X_2$. This implies that there exists a permutation $g \in Sym(n)$ such that $X_1 g = X_2$ (like above we are thinking of g acting on the edges of the graphs). Therefore $\rho_i g = \rho'_i$, and since the ρ_i's generate G, we have $(G)g = G$. Since g stabilizes G, this implies that $g \in Aut(G)$, and therefore $\mathcal{P}$ and $\mathcal{P}'$ are isomorphic polytopes. $\square$

Remark 4.2. Observe that in Theorem 4.1, we asked for G and $Aut(G)$ to be embedded as subgroups of $Sym(n)$. This is needed for instance, for the group $Sym(6)$ that has an outer automorphism. Indeed, the 5-simplex can be either generated by the five transpositions

$$(1,2),(2,3),(3,4),(4,5),(5,6)$$

or by the five 3-transpositions

$$(1,4)(2,3)(5,6),(1,2)(3,6)(4,5),(1,3)(2,4)(5,6),$$
$$(1,6)(2,3)(4,5),(1,2)(3,4)(5,6).$$

In this case, the corresponding CPR graphs will not be isomorphic. It suffices then to embed $Sym(6)$ and its automorphism group $P\Gamma L(2,9)$ in $Sym(10)$ to check that the corresponding CPR graphs are indeed isomorphic.

To relate isomorphisms of a polytope to isomorphisms of its CPR graph as in the theorem, we must embed both the group G and $Aut(G)$ into a symmetric group on the same number of points. This is always possible by embedding into $Sym_{|G|}$, but this is not practical as the CPR graphs would be large and testing for isomorphism of the graphs would be challenging.

For the efficiency test in the next sections, we embed G and $Aut(G)$ into $Sym(n)$ with n as small as possible. We give two notable examples to see how this is done.

Example 4.3. Let G be the symmetry group of the regular icosahedron. The icosahedral group G is isomorphic to $Alt(5)\times C_2$, and the automorphism group of G is isomorphic to $Sym(5)$. Thus to proceed as in the theorem, we need to embed both $Alt(5) \times C_2$ and $Sym(5)$ into $Sym(n)$ for some n. This can be done by embedding both groups into $Sym(10)$ as follows.

Let G be the subgroup of $Sym(10)$ generated by $(1,2,3)(6,7,8)$, $(3,4,5)(8,9,10)$, and $(1,6)(2,7)(3,8)(4,9)(5,10)$, and let $Aut(G)$ be the subgroup of $Sym(10)$ generated by $(1,2,3)(6,7,8)$, $(3,4,5)(8,9,10)$, and $(1,2)(6,7)$. Then we can test for isomorphism of polytopes in this group by considering isomorphisms of graphs with 10 vertices.

Example 4.4. Let G be $Sym(6)$ and $Aut(G) \cong Sym(6) \rtimes C_2$ where C_2 is generated by an odd permutation. To embed both G and $Aut(G)$ into a small $Sym(n)$ we use the isomorphisms $G \cong P\Sigma L(2,9)$ and $Aut(G) \cong P\Gamma L(2,9)$, which both naturally embed into $Sym(10)$. Thus we can classify polytopes for $Sym(6)$ by considering graphs with 10 vertices.

We point out that a nondeterministic version of our algorithm still works if we embed G into $Sym(n)$, but do not embed $Aut(G)$ into the same $Sym(n)$. For example if we tried to classify all polytopes for $G = Sym(6)$ by considering CPR graphs with only 6 vertices, we would get two non-isomorphic CPR graphs that come from the same isomorphism class of polytopes. Thus we cannot hope to completely classify polytopes up to isomorphism this way. However, we can still use the algorithm to reduce the number of isomorphism classes of polytopes considerably.

5. Comparison of Algorithms

In this section, we present two possible rewrites of the old search algorithm, one depth-first search (which we denote by the letter 'D' in Table 1) and one breadth-first search (denoted by the letter 'B' in Table 1), and compare their efficiencies to the adjusted old algorithm of "Method 2". These algorithms were given by Hartley and Hulpke in [9].

In both of the algorithms, we will be concerned with classifying all regular polytopes with automorphism group G up to isomorphism. In other words, we will be looking for all nonisomorphic ways of representing G as a string C-group $\langle \rho_0, \ldots, \rho_r \rangle$.

Both the breadth search and the depth search start out the same way.

Step 0: Find the automorphism group of G. Represent this group $Aut(G)$ as a permutation group acting on the set L of all involutions of G, as these are the possible generators in the string C group. Construct a list L_0 consisting of one representative of each orbit under the action of $Aut(G)$ on L. This gives the candidates for ρ_0.

5.1. Breadth-first algorithm for classifying polytopes.

(1) Given L_k construct L_{k+1} as follows. Let K_t be the stabilizer of a $(k+1)$-tuple $t = [r_0, \ldots, r_k]$ in L_k under the action of $Aut(G)$.
(2) Pick a set R of representatives of the orbits of the action of K_t on L.
(3) For each element r of R, check if the group $\langle t, r \rangle := \langle r_0, \ldots, r_k, r \rangle$ is a string C-group.
   (a) If it is a string C group and generates the whole group, add it to the list P.
   (b) If it is a string C-group but generates a proper subgroup of G, then add the $(k+2)$-tuple $[r_0, \ldots, r_k, r]$ to the set L_{k+1}.
(4) Stop when L_{k+1} is empty.

5.2. Depth-first algorithm for classifying polytopes. This algorithm gives a recursive approach for finding all polytopes up to isomorphism.

(1) For each element $t_i = [r_0, \ldots, r_i]$ of L_i, find the stabilizer S_{t_i} of t_i under the action of $S_{[r_0, \ldots, r_{i-1}]}$ on L.
(2) Construct a list R_{t,r_i} of representatives of the orbits of this action. This will give you candidates for ρ_{i+1}.
(3) For each element r_{i+1} of R_{t,r_i}, check if the group $\langle t, r_{i+1} \rangle$ is a string C-group.
   (a) If it is a string C group and generates the whole group then add it to the list P.

(b) If it is a string C-group but generates a proper subgroup of G then repeat the algorithm on $t_{i+1} = [r_0, \ldots, r_i, r_{i+1}]$.

5.3. **Comparison of algorithms.** In general the depth-first search algorithm has two clear advantages. First, we saw in the breadth-first search algorithm that the entire list L_1 is constructed, then the entire list L_2 is constructed before calculating L_3, and so on. These lists can be very large, so storing them can become very memory intensive for large groups. For this reason, the depth first search is usually less memory intensive.

Second, in the breadth-first search algorithm, you are computing the stabilizer of a tuple of elements $t_k = [r_0, \ldots, r_k]$ under the action of a large group $Aut(G)$. By comparison, in the depth-first search algorithm, you simply consider the stabilizer of the r_k under the action of a much smaller group which is $S_{t_{i-1}}$ which you have stored from the previous step. These two approaches yield the same stabilizing subgroup, but dealing with the action of a smaller group will usually make the depth first search faster.

We also note that there are many more ways that could be used to improve the memory use of our algorithms. For example, currently we consider $Aut(G)$ as a permutation group acting on the list L of all involutions. Instead of acting on L, one could instead view $Aut(G)$ as acting on the set of indices of each involution in this list. It is easy to convert back and forth from an index to an involution, and storing and working with these indices can be more efficient than working in the original group. This approach was in fact used in [12].

We conclude this section by giving a quantitative comparison of the 3 algorithms in the form of the following table comparing time and memory usage in the classification of all non-isomorphic polytopes for a given group.

Group	Time 2	Time B	Time D	Memory 2	Memory B	Memory D
$Alt(5)$	0.18s	0.16s	0.17s	0.01MB	0.01MB	0.01MB
$Alt(5) \times C_2$	0.29s	0.19s	0.17s	0.01MB	0.01MB	0.01MB
$P\Gamma L(2,9)$	1.86s	0.35s	0.29s	0.01MB	0.01MB	0.01MB
$Sym(7)$	5.95s	0.66s	0.45s	1.04MB	0.01MB	0.01MB
$PSL(2,25)$	0.39 s	0.27s	0.22s	0.01MB	0.01MB	0.01MB
$P\Sigma U(3,3)$	4.62 s	0.63s	0.35s	1.29MB	1.29MB	1.29MB
$PGL(2,27)$	4.38s	1.27s	0.36s	1.04MB	1.04MB	0.01MB
$Sz(8)$	0.44s	0.26s	0.26s	1.29MB	1.29MB	1.29MB
M_{12}	12.19s	1.49s	1.02s	2.32MB	2.32MB	1.29MB
J_1	54.36s	9.04s	2.78s	3.95MB	27.32MB	4.82MB
$Alt(9)$	37.12s	2.86s	1.68s	2.32MB	1.82MB	0.01MB
$Sym(9)$	2193.75s	49.42s	26.77s	10.26MB	4.54MB	1.45MB
$Alt(10)$	851.78s	41.53s	19.57s	16.36MB	8.07MB	1.57MB
HS	-	301.86s	97.75s	-	90.45MB	16.39MB

TABLE 1. Comparison of algorithms

6. Polytopes for the alternating groups

In [12], all polytopes were classified for the alternating groups A_n with $n \leq 9$. Using the algorithms which helped create the atlas [12], conjectures about the structure of possible regular polytopes for the alternating groups led to obtaining the classification of such regular polytopes for the group A_{10} as well. In order to understand larger alternating groups, the algorithms discussed in this paper were implemented. The result was the classification of all regular polytopes with automorphism group A_{11}, A_{12}, A_{13}, and A_{14}. The number of regular polytopes of each rank (up to isomorphism) is summarized in Table 2.

Group	Rank	Number of polytopes
A_{11}	3	122
A_{11}	6	4
A_{12}	3	375
A_{12}	4	173
A_{12}	5	43
A_{13}	3	3021
A_{13}	4	202
A_{13}	5	50
A_{13}	6	20
A_{14}	3	8531
A_{14}	4	255
A_{14}	5	90
A_{14}	6	18

Table 2. Number of polytopes for A_n, with $11 \leq n \leq 14$

Remark 6.1. Two nice observations come out of this data. First, the group A_{11} is the only group (of any kind) known to the authors with the property that it has regular polytopes of at least two different ranks i, j with $|i-j| \geq 2$, but is not the automorphism group of any regular polytope of rank k with $i < k < j$. On the contrary, in [4] it is proven that no symmetric group has this property. Second, for $11 < n < 15$, the group A_n has polytopes of each rank $3 \leq r \leq \lfloor \frac{n-1}{2} \rfloor := M_n$. In [5], we have proven the existence of the polytopes of rank M_n for A_n, for all $n > 11$, and we conjecture that this is in fact the maximal rank, and that regular polytopes of rank d exist for all $3 \leq d \leq M_n$.

7. Polytopes for the sporadic group Co_3

In [9], all polytopes were classified for the sporadic groups with no more than 4,030,387,200 elements; this computation was done using algorithms written in GAP. Using our programs independently written in Magma, we were able to verify the results of Hartley and Hulpke for all such sporadic groups. Also, we succeeded in classifying all polytopes for Conway's third group Co_3 of order 495,766,656,000. In this section we summarize the Schläfli types of all 22908 possible regular polytopes for this group. Note that this group has polytopes of rank at most 5.

7.1. **3-polytopes.** There are 21118 3-polytopes of type $\{p, q\}$ summarized in the following table with p giving the row and q giving the column.

	3	4	5	6	7	8	9	10	12	14	15	18	21	24	30
3	0	0	0	0	0	0	1	2	2	3	4	1	1	4	2
4	0	0	0	0	0	0	4	2	2	2	11	11	8	6	8
5	0	0	0	7	3	14	13	24	26	16	43	22	23	40	21
6	0	0	7	76	30	82	70	177	125	125	246	130	147	266	95
7	0	0	3	30	18	26	29	53	47	26	81	60	57	84	52
8	0	0	14	82	26	72	99	160	156	128	291	191	188	322	143
9	1	4	13	70	29	99	34	109	89	47	117	58	56	104	54
10	2	2	24	177	53	160	109	276	203	138	344	204	221	362	155
12	2	2	26	125	47	156	89	203	192	123	282	175	162	304	138
14	3	2	16	125	26	128	47	138	123	58	145	84	81	168	66
15	4	11	43	246	81	291	117	344	282	145	329	185	183	352	171
18	1	11	22	130	60	191	58	204	175	84	185	97	102	180	96
21	1	8	23	147	57	188	56	221	162	81	183	102	100	194	97
24	4	6	40	266	84	322	104	362	304	168	352	180	194	326	172
30	2	8	21	95	52	143	54	155	138	66	171	96	97	172	74

7.2. **4-polytopes.** There are 1746 4-polytopes of type $\{p, q, r\}$ summarized in the following nine tables, one for each value of p, with q giving the row and r giving the column.

$p = 3$	3	4	5	6	7	8	9	10	12
4	0	0	0	0	1	2	0	0	0
5	0	0	0	1	0	3	0	0	0
6	0	1	4	13	2	22	1	2	8
7	0	1	0	0	0	3	0	2	0
8	2	7	4	18	5	19	2	4	10
9	0	1	0	0	0	1	0	0	0
10	2	6	1	12	3	14	0	1	5
12	0	3	5	13	1	3	0	0	1
14	0	2	0	0	0	5	0	3	0

$p=4$	3	4	5	6	7	8	9	10	12
3	0	0	0	0	0	2	0	2	0
4	0	0	0	2	2	4	0	1	2
5	0	0	0	7	0	8	0	0	0
6	1	0	11	27	7	40	2	2	3
7	1	0	0	0	0	0	0	0	0
8	7	6	8	29	9	24	1	1	9
9	1	0	0	0	0	0	0	0	0
10	6	2	3	8	2	9	0	0	0
12	3	4	6	11	2	7	0	0	0
14	2	0	0	0	0	0	0	0	0

$p=5$	3	4	5	6	7	8	9	12
4	0	0	0	1	1	2	0	2
5	0	0	0	3	0	2	0	0
6	4	11	6	21	6	14	1	5
8	4	8	2	14	1	6	2	3
10	1	3	0	1	1	5	0	4
12	5	6	0	5	1	0	0	0

$p=6$	3	4	5	6	7	8	9	10	12
3	0	0	0	0	0	1	0	1	0
4	0	2	1	4	3	5	2	0	3
5	1	7	3	18	1	4	1	0	1
6	13	27	21	88	7	57	4	4	9
8	18	29	14	42	3	30	3	0	5
9	0	0	0	0	0	0	0	1	0
10	12	8	1	6	1	5	0	0	1
12	13	11	5	16	0	9	0	2	1

$p=7$	3	4	5	6	8	10
4	1	2	1	3	1	0
5	0	0	0	1	0	0
6	2	7	6	7	0	1
8	5	9	1	3	1	0
10	3	2	1	1	0	0
12	1	2	1	0	0	0

$p = 8$	3	4	5	6	7	8	9	10	12
3	0	2	0	1	0	2	0	1	0
4	2	4	2	5	1	6	0	1	1
5	3	8	2	4	0	0	0	2	0
6	22	40	14	57	0	26	1	5	1
7	3	0	0	0	0	0	1	0	0
8	19	24	6	30	1	8	0	0	1
9	1	0	0	0	0	0	0	0	0
10	14	9	5	5	0	4	0	0	0
12	3	7	0	9	0	0	0	0	0
14	5	0	0	0	0	0	0	0	0

$p = 9$	3	4	5	6	8	10
3	0	0	0	0	0	1
4	0	0	0	2	0	0
5	0	0	0	1	0	0
6	1	2	1	4	1	0
7	0	0	0	0	1	0
8	2	1	2	3	0	0
14	0	0	0	0	0	1

$p = 10$	3	4	6	7	8	9	12
3	0	2	1	0	1	1	0
4	0	1	0	0	1	0	0
5	0	0	0	0	2	0	0
6	2	2	4	1	5	0	1
7	2	0	0	0	0	0	0
8	4	1	0	0	0	0	0
9	0	0	1	0	0	0	0
10	1	0	0	0	0	0	0
12	0	0	2	0	0	0	0
14	3	0	0	0	0	1	0

$p = 12$	3	4	5	6	8	10
4	0	2	2	3	1	0
5	0	0	0	1	0	0
6	8	3	5	9	1	1
8	10	9	3	5	1	0
10	5	0	4	1	0	0
12	1	0	0	1	0	0

7.3. **5-polytopes.** There are 44 5-polytopes of type $\{p, q, r, s\}$ summarized in the following table.

$[p, q, r, s]$	count	$[p, q, r, s]$	count	$[p, q, r, s]$	count
$[3, 5, 6, 7]$	1	$[3, 8, 4, 6]$	1	$[5, 5, 6, 3]$	1
$[3, 6, 3, 5]$	1	$[3, 8, 6, 3]$	4	$[5, 6, 6, 3]$	1
$[3, 6, 4, 5]$	1	$[3, 8, 6, 5]$	1	$[5, 6, 8, 3]$	1
$[3, 6, 4, 7]$	1	$[3, 10, 3, 6]$	1	$[5, 8, 6, 3]$	1
$[3, 6, 5, 5]$	1	$[3, 10, 6, 3]$	1	$[6, 3, 10, 3]$	1
$[3, 6, 6, 5]$	1	$[4, 4, 6, 4]$	2	$[6, 4, 7, 3]$	1
$[3, 6, 6, 7]$	1	$[4, 4, 8, 4]$	2	$[6, 4, 8, 3]$	1
$[3, 6, 8, 3]$	4	$[4, 6, 4, 4]$	2	$[7, 4, 6, 3]$	1
$[3, 6, 8, 5]$	1	$[4, 8, 4, 4]$	2	$[7, 6, 5, 3]$	1
$[3, 6, 10, 3]$	1	$[5, 3, 6, 3]$	1	$[7, 6, 6, 3]$	1
$[3, 7, 4, 6]$	1	$[5, 4, 6, 3]$	1		
$[3, 8, 4, 5]$	1	$[5, 4, 8, 3]$	1		

8. Future work

We conclude this paper with two points about future work. First, there are other approaches being developed to be used in these classification type problems. While we build the string C-group in a "linear" fashion, by constructing the generators in their natural order, there is some hope to an improved algorithm that uses the commuting property of generators at distance at least 2, in order to restrict to analyzing fewer cases than our current algorithms. Constructing the string C-group representation in this non-linear way, could potentially lead to understanding the structure of new groups. However, the work on this is out of the scope of this paper.

Second, due to the structure of our algorithms, one can see that they would work very naturally with parallel computing. As the search trees split in these searches, there is no need to communicate between branches of the tree, so it would be natural to allow certain paths to be taken by different processors. This approach could be a necessary idea to understand the string C-group representations of some of the much larger groups.

9. Acknowledgements

We gratefully acknowledge the "Fonds David et Alice Van Buuren" and the "Communauté Française de Belgique, Action de Recherche Concertée" for their financial support with this project. We also thank an anonymous referee for useful comments and suggestions to improve the reading of the paper.

References

1. W. Bosma, J. Cannon, and C. Playoust, *The magma algebra system i: The user language*, J. Symbolic Comput. **24** (1997), 235–265.
2. P. Brooksbank and D. Vicinsky, *Three-dimensional classical groups acting on polytopes*, Discrete Comput. Geom. **44** (2010), no. 3, 654–659.
3. F. Buekenhout, *Diagrams for geometries and groups*, Journal of Combinatorial Theory, Series A **27** (1979), 121–151.
4. M. E. Fernandes and D. Leemans, *Polytopes of high rank for the symmetric groups*, Advances in Mathematics **228**, no. 6, 3207–3222.
5. M. E. Fernandes, D. Leemans, and M. Mixer, *All alternating groups A_n, with $n \geq 12$, have polytopes of rank* $\lfloor \frac{n-1}{2} \rfloor$, SIAM Journal on Discrete Math. **26** (2012), no. 2, 482–498.
6. ———, *Polytopes of high rank for the alternating groups*, Journal of Combinatorial Theory, Series A **119** (2012), no. 1, 42–56.
7. The GAP Group, *Gap – groups, algorithms and programming*, 2012, Version 4.5.6.
8. M. I. Hartley, *The atlas of small regular abstract polytopes*, Period. Math. Hungar. **53** (2006), no. 1–2, 149–156, http://www.abstract-polytopes.com/atlas/.
9. M. I. Hartley and A. Hulpke, *Polytopes derived from sporadic simple groups*, Contributions to Discrete Mathematics **5** (2010), no. 2, 106–118.
10. D. Leemans and E. Schulte, *Groups of type* $\mathrm{L}_2(q)$ *acting on polytopes*, Adv. Geom. **7** (2007), no. 4, 529–539.
11. ———, *Polytopes with groups of type* $\mathrm{PGL}_2(q)$, Ars Math. Contemp. **2** (2009), no. 2, 163–171.
12. D. Leemans and L. Vauthier, *An atlas of abstract regular polytopes for small groups*, Aequationes Math. **72** (2006), no. 3, 313–320, http://www.math.auckland.ac.nz/ dleemans/polytopes/.
13. P. McMullen and E. Schulte, *Abstract regular polytopes*, Encyclopedia of Mathematics and its Applications, vol. 92, Cambridge University Press, Cambridge, 2002.
14. D. Pellicer, *CPR graphs and regular polytopes*, European J. Combin. **29** (2008), no. 1, 59–71.

Department of Mathematics, University of Auckland
Private Bag 92019, Auckland, New Zealand
E-mail address: d.leemans@auckland.ac.nz

The Fields Institute
222 College Street, Toronto, ON M5T3J1, Canada
E-mail address: mmixer@fields.utoronto.ca