Arch. Math. 118 (2022), 19–25
© 2021 The Author(s)
0003-889X/22/010019-7
published online October 26, 2021
https://doi.org/10.1007/s00013-021-01670-5

Archiv der Mathematik

Primitive normalisers in quasipolynomial time

Mun See Chang and Colva M. Roney-Dougal

Abstract. The normaliser problem has as input two subgroups H and K of the symmetric group S_n, and asks for a generating set for $N_K(H)$: it is not known to have a subexponential time solution. It is proved in Roney-Dougal and Siccha (Bull Lond Math Soc 52(2):358–366, 2020) that if H is primitive, then the normaliser problem can be solved in quasipolynomial time. We show that for all subgroups H and K of S_n, in quasipolynomial time, we can decide whether $N_{\mathrm{S}_n}(H)$ is primitive, and if so, compute $N_K(H)$. Hence we reduce the question of whether one can solve the normaliser problem in quasipolynomial time to the case where the normaliser in S_n is known not to be primitive.

Mathematics Subject Classification. Primary 20-08; Secondary 20B35, 68W30.

Keywords. Permutation groups, Primitive groups, Computation.

1. Introduction. The *normaliser problem* asks for a generating set for $N_K(H)$, given subgroups K and H of S_n. In this paper, we investigate the theoretical complexity of the normaliser problem. It is shown in [13] that the problem can be solved in simply exponential time $2^{O(n)}$, but there is no known subexponential solution to the general problem, and in fact, the fastest practical algorithms all use a backtrack search whose worst-case complexity is greater than exponential. A permutation group problem $\mathcal{P}$ is said to be *quasipolynomial* if there exists a constant c such that $\mathcal{P}$ can be solved in time $2^{O(\log^c n)}$, where n is the degree of the underlying group or groups.

It is shown in [11] that if H is primitive, then the normaliser problem is quasipolynomial. In this paper, we will show that if $N_{\mathrm{S}_n}(H)$ is primitive, then the normaliser problem is quasipolynomial. Our main theorem is the following.

Theorem 1.1. *Let subgroups $H = \langle X \rangle$ and $K = \langle Y \rangle$ of S_n be given.*

1. *We can decide if $N = N_{\mathrm{S}_n}(H)$ is primitive, and if so, construct N in time $2^{O(\log^3 n)}$.*

2. *If N is primitive, then we can compute $N_K(H)$ in time $2^{O(\log^3 n)}$.*

(Throughout the paper, we shall assume all generating sets have size at most n: see Lemma 2.3.1).

In fact, we can compute $N_K(H)$ in time $2^{O(\log^3 n)}$ except when: (i) H is intransitive; or (ii) $|H| \geq n^{1+\lfloor \log n \rfloor}$ but H is not ample (see Definition 3.1); or (iii) $|H| < n^{1+\lfloor \log n \rfloor}$ but H does not have a small base or a small generating set (see Lemma 2.4). In this latter case, we can still compute $N_K(H)$ in quasipolynomial time $2^{O(\log^5 n)}$, see Proposition 2.5.

Babai in [1] gave a $2^{O(\log^c n)}$ time solution to the string isomorphism problem, and Helfgott in [3] showed that we can take $c = 3$. The setwise stabiliser problem is a special case of the string isomorphism problem, and was shown in [7] to be polynomial-time equivalent to the intersection problem. Hence to show that $N_K(H) = N \cap K$ can be computed in time $2^{O(\log^3 n)}$, it suffices to prove that N can be computed in time $2^{O(\log^3 n)}$.

In Section 2, we first present some preliminaries on permutation groups and permutation group algorithms. We then see how we can determine that certain groups H have base and generating set of size $O(\log n)$ in quasipolynomial time and prove Proposition 2.5. In Section 3, we introduce the class of ample groups and show that if H is ample, then $N = N_{\mathrm{S}_n}(H)$ can be computed in quasipolynomial time. Finally these results come together to prove Theorem 1.1.

2. Preliminaries and small groups.

This section first collects background on permutation groups and polynomial time computation and then studies small groups.

Let $G \leq \mathrm{Sym}(\Omega)$ and $H \leq \mathrm{Sym}(\Gamma)$. Then G and H are *permutation isomorphic* if there exist an isomorphism $\phi : G \to H$ and a bijection $\sigma : \Omega \to \Gamma$ such that $\sigma(\omega^g) = \sigma(\omega)^{\phi(g)}$ for all $\omega \in \Omega$ and $g \in G$. We say that such a pair (ϕ, σ) is a *permutation isomorphism* from G to H.

Notation 2.1. Let $[m]_k$ denote the set of all k-subsets of $\{1, 2, \ldots, m\}$ with $1 \leq k \leq m/2$. Let $\mathrm{A}_{m,k}$ and $\mathrm{S}_{m,k}$ denote A_m and S_m, acting on $[m]_k$. Let $[m]_k^l$ denote the set of all l-tuples of $[m]_k$, and let $A(m, k, l)$ be the group $(\mathrm{A}_{m,k})^l$ acting coordinatewise on $[m]_k^l$.

We will be using the following key result, proved by Maróti using the Classification of Finite Simple Groups.

Theorem 2.2 ([9]). *Let G be a primitive subgroup of S_n. Then at least one of the following holds.*

1. *G is M_{11}, M_{12}, M_{23}, or M_{24} with their 4-transitive actions.*
2. *There exist $m \geq 5$, $1 \leq k < m/2$, and $l \geq 1$ such that, up to permutation isomorphism, $A(m, k, l) \trianglelefteq G \leq \mathrm{S}_{m,k} \wr \mathrm{S}_l$.*
3. *$|G| < n^{1+\lfloor \log n \rfloor}$.*

We shall call these classes *Mathieu*, *large*, and *small*, respectively. A primitive group is of *type PA* if it is in product action and the component of the base group is almost simple (see [6]). It follows that a large primitive group is either almost simple (when $l = 1$) or of type PA (when $l > 1$).

For $G = \langle z_1, z_2, \ldots, z_k \rangle \leq \mathrm{S}_n$ and $L = \langle y_1, y_2, \ldots, y_l \rangle \leq \mathrm{S}_m$, a homomorphism $\phi : G \to L$ is *given by generator images* if it is encoded by a list $[z_1, \ldots, z_k, y_1, \ldots, y_l, \phi(z_1), \ldots, \phi(z_k)]$. We shall assume that all homomorphisms are given by generator images, that we have a library of standard representations of all finite simple groups, and that their automorphism groups are known.

The following results are standard (see, for example, [12, §3.1] or [4, §4]).

Lemma 2.3. *Given $G = \langle Z \rangle \leq \mathrm{S}_n$, the following can be done in time polynomial in $|Z| \cdot n$.*

1. *Replace Z by a generating set for G of size at most n; given $\sigma \in \mathrm{S}_n$, decide if $\sigma \in G$; compute $|G|$; compute the orbits of G; compute the stabiliser in G of any given point; compute an irredundant base for G; decide if G is primitive.*
2. *Given a map $\phi : G \to L$ by the images of Z, decide if ϕ extends to an isomorphism; given an isomorphism $\phi : G \to L$, compute ϕ^{-1}.*
3. *Find a minimal normal subgroup of G; compute $C_G(J)$ for $J \trianglelefteq G$; find generators for the socle* $\mathrm{soc}(G)$.
4. *Compute the composition factors of G; decide if G is simple and if so, give an isomorphism from G to a standard representation.*

Next, we show that we can find a small base and a small generating set for certain groups H in quasipolynomial time. For a group $G \leq \mathrm{S}_n$, let $d(G)$ and $b(G)$ denote the size of the smallest generating set and base of G, respectively.

Lemma 2.4. *Let $H = \langle X \rangle \leq \mathrm{S}_n$ be given.*

1. *If $|H| \leq n^{1+\lfloor \log n \rfloor}$, then in time $2^{O(\log^3 n)}$, we can decide if $d(H) \leq \lceil \log n \rceil$, and if so, output such a generating set.*
2. *In time $2^{O(\log^2 n)}$, we can decide if $b(H) \leq \lceil \log n \rceil + 1$, and if so, output such a base.*
3. *If $N = N_{\mathrm{S}_n}(H)$ is a small primitive group, then $d(H) \leq \lceil \log n \rceil$ and $b(H) \leq \lceil \log n \rceil + 1$.*

Proof. Part 1: We consider all $(\lceil \log n \rceil)$-tuples Z of elements of H and decide for each Z if $\langle Z \rangle = H$. The number of such tuples is $|H|^{\lceil \log n \rceil} \in 2^{O(\log^3 n)}$. By Lemma 2.3.1, for each such tuple Z, we can decide if $\langle Z \rangle = H$ in polynomial time.

Part 2: We consider all $(\lceil \log n \rceil + 1)$-tuples B over $\{1, 2, \ldots, n\}$. For each such B, we check if B is a base of H by checking if $H_{(B)} = 1$, which can be done in polynomial time by Lemma 2.3.1. Since there are $n^{\lceil \log n \rceil + 1} \in 2^{O(\log^2 n)}$ tuples to consider, the result follows.

Part 3: If N is a small primitive group, then H has order at most $n^{1+\lfloor \log n \rfloor}$. Since H is a normal subgroup of a primitive group, by [5, Theorem 1.1], $d(H) \leq \log n$ or $H = \mathrm{S}_3$, so $d(H) \leq \lceil \log n \rceil$. By [10], $b(H) \leq b(N) \leq \lceil \log n \rceil + 1$. □

Lastly we observe that the normaliser problem for groups of order less than $n^{1+\lfloor \log n \rfloor}$ can be solved in quasipolynomial time, even if they are not primitive.

Proposition 2.5. *Let $H = \langle X \rangle \leq \mathrm{S}_n$ be given. If $|H| < n^{1+\lfloor \log n \rfloor}$, then $N_{\mathrm{S}_n}(H)$, and hence $N_K(H)$, can be computed in time $2^{O(\log^5 n)}$.*

Proof. By Lemma 2.3.1, in polynomial time, we can check that $|H| < n^{1+\lfloor \log n \rfloor}$, compute an irredundant base B for H, and remove from $X = \{x_1, x_2, \ldots, x_s\}$ the generators x_i where $x_i \in \langle x_1, \ldots, x_{i-1} \rangle$. This gives a base B and a generating set Z for H of size at most $\log |H| \in O(\log^2 n)$.

In [11, proof of Theorem 3.3], it is shown that in time $2^{O(|Z||B| \log n)}$, we can construct a set containing all $|Z|$-tuples of elements of H that are images of Z under conjugation by elements of $N_{\mathrm{S}_n}(H)$. By Lemma 2.3.2 and [8, Lemma 3.5], for each such potential image, we can determine a conjugating element $\sigma \in N_{\mathrm{S}_n}(H)$ or show that no such σ exists in polynomial time. □

3. Ample groups. In this section, we will introduce ample groups, and show that if $N = N_{\mathrm{S}_n}(H)$ is a large primitive group, then H is ample. We then show that in quasipolynomial time, we can decide if a given group is ample and if so compute its normaliser. Finally, we will prove Theorem 1.1.

Definition 3.1. We define a subgroup H of S_n to be *ample* if there exist $m \geq 5$, $1 \leq k < m/2$, and $l \geq 1$ such that $\mathrm{soc}(H)$ is permutation isomorphic to $A(m, k, l)$.

Notice that an ample group can be imprimitive.

Lemma 3.2. *Let H be a subgroup of S_n such that $N = N_{\mathrm{S}_n}(H)$ is a large primitive group. Then $\mathrm{soc}(N) = \mathrm{soc}(H)$, and H is ample.*

Proof. We first show that $\mathrm{soc}(N) = \mathrm{soc}(H)$. The group $\mathrm{soc}(H)$ is characteristic in H, so $\mathrm{soc}(H) \trianglelefteq N$. A large primitive group is either almost simple or of type PA, and so N has a unique minimal normal subgroup (see [6, §1]). Therefore

$$\mathrm{soc}(N) \leq \mathrm{soc}(H) \leq H, \text{ so } \mathrm{soc}(N) \trianglelefteq H.$$

To see that $\mathrm{soc}(H) \leq \mathrm{soc}(N)$, let M be a minimal normal subgroup of H. Then either $M \leq \mathrm{soc}(N)$ or $M \cap \mathrm{soc}(N) = 1$. If $M \cap \mathrm{soc}(N) = 1$, then $M \leq C_N(\mathrm{soc}(N))$. But by [2, Theorem 4.3B], $C_N(\mathrm{soc}(N)) = 1$, a contradiction. Therefore, all minimal normal subgroups of H are contained in $\mathrm{soc}(N)$, hence $\mathrm{soc}(H) \leq \mathrm{soc}(N)$, and so $\mathrm{soc}(N) = \mathrm{soc}(H)$.

The largeness of N implies that there exist m, k, and l such that $\mathrm{soc}(N)$ is permutation isomorphic to $A(m, k, l)$. Now since $\mathrm{soc}(H) = \mathrm{soc}(N)$, the group H is ample. □

The following is well known (see [2, Theorem 4.5A] for example).

Lemma 3.3. *Let $W \leq \mathrm{Sym}([m]_k^l)$ be $\mathrm{S}_{m,k} \wr \mathrm{S}_l$ acting on $[m]_k^l$ for some $m \geq 5$, $1 \leq k < m/2$, and $l \geq 1$. Then the normaliser in $\mathrm{Sym}([m]_k^l)$ of $A(m, k, l)$ is W.*

Next we give a polynomial time algorithm to decide whether H is ample.

Lemma 3.4. *Given $H = \langle X \rangle \leq \mathrm{S}_n$, in polynomial time, we can decide if H is ample, and if so, output a permutation isomorphism from $\mathrm{soc}(H)$ to $A(m, k, l)$ for some m, k, and l.*

Proof. By Lemma 2.3.3, we can compute a generating set for $S := \mathrm{soc}(H)$ in polynomial time. The group S is a direct product of simple groups, so we can decide whether $S \cong \mathrm{A}_m^l$, for some $m \geq 5$ and $l \geq 1$, by checking if S has l composition factors, each isomorphic to A_m. By Lemma 2.3.4, this can be done in polynomial time.

If S is isomorphic to A_m^l, we next determine whether there exists a k such that $1 \leq k < m/2$ and $n = \binom{m}{k}^l$. If so, we construct an isomorphism $\iota : S \to A(m,k,l)$ as follows. Initialise $N_1 = S$, then for $2 \leq i \leq l$, we iteratively find a minimal normal subgroup M_i of N_i and take $N_{i+1} = C_{N_i}(M_i)$ in polynomial time by Lemma 2.3.3. Then $M_i \cong \mathrm{A}_m$ and $N_i = M_i \times C_{N_i}(M_i)$ for each i and so $S = M_1 \times M_2 \times \cdots \times M_l$. We construct an isomorphism $\iota : S \to A(m,k,l)$ using an isomorphism from each M_i to a direct factor of $A(m,k,l)$, via isomorphisms to a standard copy of A_m, in polynomial time by Lemma 2.3.4.

It remains to show how to find a permutation isomorphism between S and $A(m,k,l)$. Let $\Delta = [m]_k^l$ and let $W \leq \mathrm{Sym}(\Delta)$ be as in Lemma 3.3. If $m = 6$, then there exists an involution α such that $\mathrm{Aut}(\mathrm{A}_{m,k}) = \langle \mathrm{S}_{m,k}, \alpha \rangle$, and we can obtain such an α in polynomial time by Lemma 2.3.4. So in polynomial time, we can write down all $2^l \leq 2^{\log n} = n$ coset representatives of W in $\mathrm{Aut}(A(m,k,l))$. We check if S and $A(m,k,l)$ are permutation isomorphic by checking if there exist such a coset representative λ and a bijection $\sigma : \{1,2,\ldots,n\} \to \Delta$ such that $(\iota\lambda, \sigma)$ is a permutation isomorphism, in polynomial time by [11, Lemma 2.7]. If $m \neq 6$, then $\mathrm{Aut}(\mathrm{A}_{m,k}) = \mathrm{S}_{m,k}$ and so $\mathrm{Aut}(A(m,k,l)) = W$, and we may set $\lambda = 1$. □

If H is ample and $l = 1$, then H is almost simple. The next result considers the case where H is ample and $l > 1$.

Theorem 3.5. *Given $H = \langle X \rangle \leq \mathrm{S}_n$, we can decide if H is ample and not almost simple, and if so, compute $N = N_{\mathrm{S}_n}(H)$ in time $2^{O(\log n \log\log n)}$.*

Proof. By Lemma 3.4, in polynomial time, we can check if H is ample and not almost simple, and if so obtain a permutation isomorphism (ϕ, σ) from $\mathrm{soc}(H)$ to $A(m,k,l)$.

We first show that we can compute a generating set for $M = N_{\mathrm{S}_n}(\mathrm{soc}(H))$ of size at most four in polynomial time. Let W be as in Lemma 3.3. The bijection σ^{-1} induces an isomorphism from $\mathrm{Sym}([m]_k^l)$ to S_n that maps W to M. By [11, Lemma 4.3], we can write down a generating set Z for W of size four, so $M = \langle \sigma^{-1}(Z) \rangle$ can be computed in polynomial time by Lemma 2.3.2.

Next, since M is isomorphic to W,

$$[M : \mathrm{soc}(H)] \leq 2^l |\mathrm{S}_l| \leq 2^l l^l = 2^{l + l \log l}.$$

As $l \leq \log n$, it follows that $[M : \mathrm{soc}(H)] \leq 2^{\log n + \log n \log\log n}$. Notice that $\mathrm{soc}(H) \trianglelefteq H$, so $H \leq M$. Therefore $[M : H] \leq [M : \mathrm{soc}(H)] \leq 2^{2 \log n \log\log n}$.

By [11, Lemma 4.4], the group N can be computed in time $O(n^3[M : H^2]) = 2^{O(\log n \log\log n)}$ (it is assumed in [11] that H is primitive, but the assumption is not needed in the proof). □

We end by giving the proof of Theorem 1.1.

Proof of Theorem 1.1. We first prove Part 1. Without loss of generality, suppose that H is non-trivial. We check if H is transitive, in polynomial time by Lemma 2.3.1. If not, then $N = N_{\mathrm{S}_n}(H)$ is not primitive, and we return false.

Otherwise, by [11, Lemma 4.5], in polynomial time, we can check if H is almost simple and if so, compute N. Assume from now on that H (and hence N) is not almost simple. Next, by Theorem 3.5, in time $2^{O(\log n \log\log n)}$, we can determine if H is ample and if so, compute N.

If H is not ample, then by Theorem 3.2, N is not large. So by Theorem 2.2, N is primitive if and only if N is small. We check if $|H| \leq n^{1+\lfloor \log n \rfloor}$ in polynomial time by Lemma 2.3.1 and return false if not. Next we look for a generating set of size at most $\lceil \log n \rceil$ and a base of size at most $\lceil \log n \rceil + 1$ for H in time $2^{O(\log^3 n)}$ by Lemma 2.4.1-2. If no such base and generating set exist, then by Lemma 2.4.3, N is not primitive, and we return false. Otherwise we compute N in time $2^{O(\log^3 n)}$ by [11, Theorem 3.3], and check if N is primitive, in polynomial time by Lemma 2.3.1.

Part 2 now follows from the fact that, given N, the group $N_K(H) = K \cap N$ can be computed in time $2^{O(\log^3 n)}$ by Babai and Helfgott's results [1,3,7]. □

Acknowledgements. The first author is supported by a Royal Society Grant (RGF\EA\181005).

Open Access. This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

[1] Babai, L.: Graph isomorphism in quasipolynomial time. In: STOC'16-Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, pp. 684–697. ACM, New York (2016)

[2] Dixon, J.D., Mortimer, B.: Permutation Groups. Graduate Texts in Mathematics, vol. 163. Springer-Verlag, New York (1996)

[3] Helfgott, H.: Isomorphismes de graphes en temps quasi-polynomial (d'après Babai et Luks, Weisfeiler-Leman...). Astérisque **407**, 135–182 (2019)

[4] Holt, D.F., Eick, B., O'Brien, E.A.: Handbook of Computational Group Theory. Chapman & Hall, Boca Raton (2005)

[5] Holt, D.F., Roney-Dougal, C.M.: Minimal and random generation of permutation and matrix groups. J. Algebra **387**, 195–214 (2013)

[6] Liebeck, M.W., Praeger, C.E., Saxl, J.: On the O'Nan-Scott theorem for finite primitive permutation groups. J. Austral. Math. Soc. Ser. A **44**(3), 389–396 (1988)

[7] Luks, E.M.: Permutation groups and polynomial-time computation. In: Groups and Computation (New Brunswick, NJ, 1991), pp. 139–175. DIMACS Ser. Discrete Math. Theoret. Comput. Sci., 11. Amer. Math. Soc., Providence, RI (1993)

[8] Luks, E.M., Miyazaki, T.: Polynomial-time normalizers. Discrete Math. Theoret. Comput. Sci. **13**(4), 61–96 (2011)

[9] Maróti, A.: On the orders of primitive groups. J. Algebra **258**(2), 631–640 (2002)

[10] Moscatiello, M., Roney-Dougal, C.M.: Base sizes of primitive permutation groups. Monatsh Math. (2021). https://doi.org/10.1007/s00605-021-01599-5

[11] Roney-Dougal, C.M., Siccha, S.: Normalisers of primitive permutation groups in quasipolynomial time. Bull. Lond. Math. Soc. **52**(2), 358–366 (2020)

[12] Seress, A.: Permutation Group Algorithms. Cambridge Tracts in Mathematics, vol. 152. Cambridge University Press, Cambridge (2003)

[13] Wiebking, D.: Normalizers and permutational isomorphisms in simply-exponential time. In: Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, pp. 230–238. SIAM, Philadelphia, PA (2020)

Mun See Chang
School of Computer Science
University of St Andrews
St Andrews
UK
e-mail: msc2@st-andrews.ac.uk

Colva M. Roney-Dougal
School of Mathematics and Statistics
University of St Andrews
St Andrews
UK
e-mail: Colva.Roney-Dougal@st-andrews.ac.uk

Received: 3 June 2021

Revised: 7 September 2021

Accepted: 20 September 2021.