The complexity of the Weight Problem for permutation and matrix groups

Peter J Cameron [1]

School of Mathematical Sciences, Queen Mary, University of London, London, E1 4NS, UK

Taoyang Wu [2]

Department of Computer Science and School of Mathematical Sciences, Queen Mary, University of London, London, E1 4NS, UK

Abstract

Given a metric d on a permutation group G, the corresponding weight problem is to decide whether there exists an element $\pi \in G$ such that $d(\pi, e) = k$, for some given value k of d. In this paper we show that this problem is **NP**-complete for many well-known metrics. We also consider the problem of finding the maximum or minimum weight of an element of G.

Key words: Weight Problem, Metrics, Permutation Group, **NP**-complete

---

1 Introduction

Given a metric d on S_n, the *weight* of $\pi \in S_n$ is defined to be $w_d(\pi) = d(\pi, e)$, where e is the identity. Now we are interested in the following weight problems:

Problem 1 *d-Weight Problem*
Instance: Generators for G in the form of products of cycles, and k in the range of d.
Question: Whether there is an element $\pi \in G$ such that $w_d(\pi) = k$.

[1] Email: `P.J.Cameron@qmul.ac.uk`
[2] Email: `Taoyang.Wu@dcs.qmul.ac.uk`

Problem 2 *d-Maximum Weight Problem*
Instance: Generators for G in the form of products of cycles, and k in the range of d.
Question: Whether $\max_{\pi \in G} w_d(\pi) = k$.

Problem 3 *d-Minimum Weight Problem*
Instance: Generators for G in the form of products of cycles, and k in the range of d.
Question: Whether $\min_{\pi \in G \setminus \{e\}} w_d(\pi) = k$.

Often the permutation group G is given by a set of generating permutations $\{g_1, g_2, \cdots, g_m\}$ where each g_i is presented as the product of cycles. From such input much information, such as $|G|$ and a membership test, can be obtained by the Schreier–Sims algorithm in polynomial time [5]. There are also many other polynomial algorithms obtained for different properties of G. For further information, [17] is a good resource.

Despite the discovery of these polynomial algorithms, many complexity issues concerning G remain unknown. The main result of this paper is that the weight problem of many well known metrics (Section 2) is **NP**-complete.

The **NP**-completeness of the weight problem for the Hamming metric was independently discovered by Buchheim and Jünger in [2]. In fact, the **NP**-completeness of the weight problem for linear binary codes was proved by Berlekamp *et al.* in 1978 [1], and we give a simple reduction from the permutation group problem to the coding problem. On the other hand, the complexity of the subgroup distance problem was discussed by Pinch for the Cayley metric in [15], and later generalized to other cases by Buchheim *et al.* in [4].

For the remainder of this paper, we will survey some metrics on a permutation group in Section 2. The **NP**-completeness of the weight problem and minimum weight problem for the Hamming metric and most of those defined in Section 2 is derived from the reduction to linear codes in Section 3. The maximum weight problem for Hamming metric is proved in Section 4, where we show that deciding whether G contains fixed-point-free elements is **NP**-complete. The result for most of the other metrics is given in Section 5. The case l_∞ is a bit different, since the maximum weight problem is in **P**; the other two problems are shown to be **NP**-complete in Section 6. In Section 7, we consider some related questions, including the complexity of these questions for transitive groups. Finally, in Section 8 we conclude with some open problems.

2 Some metrics on permutation groups

A metric d on S_n is called *right-invariant* if $d(\pi,\sigma)=d(\pi\tau,\sigma\tau)$ for any $\pi,\sigma\tau\in S_n$. If d is right-invariant, then $d(\pi,\sigma)=d(\pi\sigma^{-1},e)=w_d(\pi\sigma^{-1})$. Conversely, if w is any function from S_n to the non-negative real numbers satisfying

- $w(\pi)=0$ if and only if $\pi=e$,
- $w(\pi\sigma)\le w(\pi)+w(\sigma)$ for any $\pi,\sigma\in S_n$,

then w is the norm derived from a right-invariant metric on S_n.

A metric d is *left-invariant* if $d(\pi,\sigma)=d(\tau\pi,\tau\sigma)$ for any $\pi,\sigma,\tau\in S_n$. If w_d is the norm derived from a right-invariant metric d, then d is left-invariant if and only if w_d is *conjugation-invariant*, that is, $w(\sigma\pi\sigma^{-1})=w(\pi)$ for all $\pi,\sigma\in S_n$. This holds if and only if the metric d does not depend on the ordering of the set $\{1,\ldots,n\}$ on which S_n acts.

In this section we will survey some well known right-invariant metrics on S_n. For more detailed discussion, we recommend [9,8].

- *Hamming Distance*: $H(\pi,\sigma)=|\{i|\pi(i)\neq\sigma(i)\}|$.
- *Cayley Distance*: $T(\pi,\sigma)$ is the minimum number of transpositions taking π to σ.
- *Movement*: The movement of a permutation π (see [16]) is defined as

$$M(\pi)=\max_{A\subseteq\{1,\ldots,n\}}|\pi(A)\setminus A|.$$

  This is easily seen to be a norm, and so the corresponding metric given by $d_M(\pi,\sigma)=M(\sigma\pi^{-1})$ is right-invariant.
- *Footrule*: $l_1(\pi,\sigma)=\sum_{i=1}^{n}|\pi(i)-\sigma(i)|$.
- *Spearman's rank correlation*: $l_2(\pi,\sigma)=\sqrt{\sum_{i=1}^{n}(\pi(i)-\sigma(i))^2}$.
- l_p $(1\le p\le\infty)$: $l_p(\pi,\sigma)=\sqrt[p]{\sum_{i=1}^{n}(\pi(i)-\sigma(i))^p}$.
- $l_\infty(\pi,\sigma)=\max_{1\le i\le n}|\pi(i)-\sigma(i)|$.
- *Lee Distance*: $L(\pi,\sigma)=\sum_{i=1}^{n}\min(|\pi(i)-\sigma(i)|,n-|\pi(i)-\sigma(i)|)$
- *Kendall's tau*: $I(\pi,\sigma)=$ the minimum number of pairwise adjacent transpositions needed to obtain σ from π, i.e,

$$I(\pi,\sigma)=|\{(i,j)|1\le i,j\le n,\pi(i)<\pi(j),\sigma(i)>\sigma(j)\}.|$$

- *Ulam's Distance*: $U(\pi,\sigma)=n-k$, where k is the length of the longest increasing subsequence in $(\sigma\pi^{-1}(1),\cdots,\sigma\pi^{-1}(n))$.

For a taste of the above metrics, we give the following table to show the weight of elements in Klein four-group $G=\langle(1,2)(3,4)\ (1,3)(2,4)\rangle$.

π	cycles	H	M	T	l_1	l_2	l_p	l_∞	L	I	U
$[1,2,3,4]$	(1)(2)(3)(4)	0	0	0	0	0	0	0	0	0	0
$[2,1,4,3]$	(1,2)(3,4)	4	2	2	4	2	$\sqrt[p]{4}$	1	4	2	2
$[3,4,1,2]$	(1,3)(2,4)	4	2	2	8	4	$2\sqrt[p]{4}$	2	8	4	2
$[4,3,2,1]$	(1,4)(2,3)	4	2	2	8	$\sqrt{20}$	$\sqrt[p]{2(1+3^p)}$	3	4	6	3

The Hamming, Cayley and movement metrics are left-invariant; the others are not.

We note that the minimum Hamming weight of a permutation group G is usually called the *minimal degree* of G; this parameter was extensively studied in the classical literature of permutation groups.

3 A connection with coding theory

Recall that the Hamming weight $w(c)$ of a binary word c of length n is defined to be the number of non-zero coordinates of c. A linear binary code C is a subspace of $\mathbb{F}_2^n$, given by a set of words forming a basis for C. The Hamming weight and maximum and minimum weight problems for linear codes are defined as in the permutation group case.

Given a linear binary code C of length n, we can construct a permutation group $G(C) \leq S_{2n}$, isomorphic to the additive group of C, as follows: to each codeword $c \in C$, we associate a permutation π_c which interchanges $2i-1$ and $2i$ if $c_i = 1$, and fixes these two points if $c_i = 0$. Now the Hamming weight of π_c is twice the Hamming weight $w(c)$ of c. Moreover, since $|\pi_c(i) - i| \leq 1$ for all i, the weights defined by all our metrics except l_∞ are monotonic functions of the Hamming weight of c: we have

- $w_T(\pi_c) = w_M(\pi_c) = w_I(\pi_c) = w_U(\pi_c) = w(c)$;
- $w_H(\pi_c) = w_L(\pi_c) = 2w(c)$;
- $w_{l_p}(\pi_c) = (2w(c))^{1/p}$.

Berlekamp *et al.* [1] proved that the weight problem for linear binary codes is **NP**-complete, and their method was adapted by Vardy [19] to show that the minimum weight problem for linear codes is also **NP**-complete. It follows that the weight problem and minimum weight problem for all the metrics given in Section 2 except for l_∞ are **NP**-complete:

Theorem 4 *The weight problem and the minimum weight problem for the Hamming, Cayley, movement, l_p (for $1 \leq p < \infty$), Kendall's tau, Lee and*

Ulam metrics are all **NP**-*complete.*

The weight and minimum weight problem for l_∞ requires separate treatment, as does the maximum weight problem for all metrics.

4 Maximum Hamming weight problem and the FPF problem

The largest possible weight for a linear code C of length n is n; this is attained only if C contains the all-1 vector. Given a basis for C, this can be checked in polynomial time. So we need a different argument for the maximum weight problem.

Elements $g \in G$ with Hamming weight n (also called *fixed point free* elements or *derangements* are of special interest in many applications. Formally, we have

$$W_H(g) = n \Leftrightarrow \mathrm{fix}_\Omega(g) = \{\alpha \in \Omega | \alpha g = \alpha\} = \emptyset.$$

All such elements form a subset of G, denoted by:

$$\mathrm{FPF}(G) = \{g | W_H(g) = n\} = \{g \in G | (\forall \alpha \in \Omega) \alpha g \neq \alpha\}.$$

In short, we will call G fixed point free (**FPF**) if $\mathrm{FPF}(G) \neq \emptyset$. Therefore, the problem of deciding whether there is an element $g \in G$ with Hamming weight n is the the same as the following problem:

Problem 5 *Fixed-Point-Free (***FPF***)*
Instance: Generators for G in the form of product cycles.
Question: Whether G is **FPF**.

Since we can verify whether or not $g \in \mathrm{FPF}(G)$ in polynomial time by checking the action of g on each point of Ω, **FPF** belongs to **NP**. Now we will prove the NP-completeness of the maximum Hamming weight problem by showing that **FPF** is NP-complete. To this end, we constructs a polynomial-time reduction from **NAESAT**, a **NP**-complete problem [14] defined as:

Problem 6 NAESAT
Instance: Collection $C = \{c_1, c_2, \cdots, c_m\}$ of clauses on a finite set U of boolean variables such that $|c_i| = 3$ for $1 \leq i \leq m$.
Question: Is there a truth assignment for U such that in no clause are all three literals equal in truth value (neither all true nor all false)?

Given an arbitrary instance of **NAESAT** (U, C), that is, a set of clauses $C = \{c_1, c_2, \cdots, c_m\}$ each with three literals, involving the variables $x_1, \cdots, x_n$, we will construct a permutation group G such that G is **FPF** if and only if there is a truth assignment on the variables such that no clause has all literals true, or all literals false.

G is generated by $2n$ generators $\{g_1, g'_1, \cdots, g_n, g'_n\}$ where the cycles structure of each generator is given as follows.

Step 1: For each x_i in U, we have the variable gadget $(2i-1, 2i)$ and associate it with generators g_i and g'_i.

Step 2: For each clause $C_j = c_{j,1} \vee c_{j,2} \vee c_{j,3}$, we have the clause gadgets

$$\begin{aligned} h_{j,1} &= (p+1, p+2)(p+3, p+4) \\ h_{j,2} &= (p+1, p+3)(p+2, p+4) \\ h_{j,3} &= (p+1, p+4)(p+2, p+3) \end{aligned}$$

where $p = 2n + 4(j-1)$.

Each clause gadget is associated with a generator via the following way:

- If $c_{j,k} = x_t$, then $h_{j,k}$ is associated with generator g_t.
- If $c_{j,k} = \bar{x}_t$, then $h_{j,k}$ is associated with generator g'_t.

To show how the above transformation works, we give an example:

Example: The transformation from **NAESAT** to **FPF**.
We are given an instance of **NAESAT** (U, C) as follows: $U = \{x_1, x_2, x_3\}$, and $C = \{c_1, c_2, c_3, c_4\}$, where
$c_1 = x_1 \vee x_2 \vee x_3$,
$c_2 = \bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3$,
$c_3 = x_1 \vee x_2 \vee \bar{x}_3$,
$c_4 = \bar{x}_1 \vee \ x_2 \vee x_3$.
Then $\{t(x_1) = \mathrm{T},\ t(x_2) = \mathrm{F},\ t(x_3) = \mathrm{T}\}$ is a satisfying truth assignment such that all clauses take diverse values.

By the above transformation process, we have $G = \langle g_1, g'_1, g_2, g'_2, g_3, g'_3 \rangle$, which acts on $\Omega = \{1, 2, \cdots, 22\}$.

- $g_1 = (1,2)h_{1,1}h_{3,1} = (1,2)(7,8)(9,10)(15,16)(17,18)$;
- $g'_1 = (1,2)h_{2,1}h_{4,1} = (1,2)(11,12)(13,14)(19,20)(21,22)$;
- $g_2 = (3,4)h_{1,2}h_{3,2}h_{4,2} = (3,4)(7,9)(8,10)(15,17)(16,18)(19,21)(20,22)$;
- $g'_2 = (3,4)h_{2,2} = (3,4)(11,13)(12,14)$;
- $g_3 = (5,6)h_{1,3}h_{4,3} = (5,6)(7,10)(8,9)(19,22)(20,21)$;

- $g'_3 = (5,6)h_{2,3}h_{3,3} = (5,6)(11,14)(12,13)(15,18)(16,17)$.

It's straightforward to show that $g = g_1 g'_2 g_3 \in \text{FPF}(G)$, corresponding to the truth assignment.

Because each instance of **NAESAT** with n variables and m clauses will be transformed to a group with $2n$ generators acting on a set with $2n+4m$ points, such procedure can be completed in polynomial time. Now we claim:

Lemma 7 $\text{FPF}(G) \neq \emptyset$ *if and only if* (U, C) *has a truth assignment such that each clause has diverse values.*

PROOF.

Suppose t is a truth assignment of (U, C) satisfying the condition in the lemma. We want to show that

$$g = g_1^{y_1} {g'_1}^{1-y_1} \cdots g_n^{y_n} {g'_n}^{1-y_n} \in \text{FPF}(G),$$

where $y_j = 0$ if $t(x_1) = \text{F}$ and $y_j = 1$ otherwise.

For each $\alpha \in \Omega$, it belongs to one of the following two cases:

- $\alpha \leq 2n$. This implies $\exists 1 \leq i \leq n \ s.t \ \alpha \in [2i-1, 2i]$, therefore $\alpha g = \alpha g_i^{y_i} {g'_i}^{1-y_i} = \alpha(2i-1, 2i)^{y_i+1-y_i} = \alpha(2i-1, 2i) \neq \alpha$.
- $\alpha > 2n$. This means $\alpha \in [p+1, p+4]$ for some $p = 2n+4(k-1),\ 1 \leq k \leq m$. W.l.o.g, we can assume $\alpha = 2n+1$ and $c_1 = x_1 \vee x_2 \vee x_3$. Then

  $$\alpha g = \alpha g_1^{y_1} g_2^{y_2} g_3^{y_3} = \alpha h_{1,1}^{y_1} h_{1,2}^{y_2} h_{1,3}^{y_3} \neq \alpha$$

  because $v = (y_1, y_2, y_3) \neq (0,0,0)$ and $v \neq (1,1,1)$.

Similarly, if $g \in \text{FPF}(G)$, then g can be expressed as $g_1^{y_1} {g'_1}^{z_1} \cdots g_n^{y_n} {g'_n}^{z_n}$ for some $(y_1, \cdots, y_n, z_1, \ldots, z_n)$ where $y_k, z_k \in \{0, 1\}$ because each generator is of order 2. If g is FPF on $\{2i-1, 2i\}$ for $i \leq n$, we have $z_i = 1 - y_i$. Then the fact that g is FPF on $\{2n+1, \ldots, 2n+4m\}$ shows that the truth assignment t corresponding to $(y_1, \cdots, y_n)$ satisfies the condition of **NAESAT**.

The above lemma implies:

Theorem 8 **FPF** *is* **NP**-*complete.*

Our construction shows more:

Corollary 9 **FPF** *is* **NP**-*complete even when* G *is an elementary abelian*

2-*group and each orbit has size at most* 4.

Because **FPF** is a special case of the maximum Hamming weight problem, now we obtain the following theorem:

Theorem 10 *The maximum Hamming weight problem is* **NP***-complete, even when G is an elementary abelian* 2*-group and each orbit has size at most* 4.

We remark that the **NP**-completeness of **FPF** can be proved by a reduction from **3SAT**, as was done in [2], but the argument given here allows smaller groups to be used. This will be important for similar arguments in Section 7.4.

5 The maximum weight problem for other metrics

In this section we will consider the maximum weight problems corresponding to the metrics defined in Section 2, with the exception of l_∞.

5.1 Cayley weight problem and movement

Lemma 11 *For an elementary abelian* 2*-group G, we have*

$$W_H(g) = M(g) = 2 \cdot W_T(g) \text{ for all } g \in G.$$

PROOF. Because G is an elementary abelian 2-group, we know each $g \in G$ has only 1-cycles and 2-cycles. And any 1-cycle contributes 0 to Hamming and Cayley weights and movement, while 2-cycles contribute 2 to the Hamming weight and 1 to the Cayley weight and movement.

Theorem 12 *The movement and Cayley weight problems are* **NP***-complete, even when G is an elementary abelian* 2*-group and each orbit has size at most* 4.

5.2 l_p weight problem

We define the *span* of an orbit O to be $\max(O) - \min(O)$.

We need to modify the clause gadgets a bit to get the transformation from **NAESAT**: for each clause $C_j = c_{j,1} \vee c_{j,2} \vee c_{j,3}$, the clause gadgets should be:

$$
\begin{aligned}
h'_{j,1} = {} & (q+1, q+2)(q+3, q+4)(q+5, q+7)(q+6)(q+8) \\
& (q+9, q+12)(q+10, q+11) \\
h'_{j,2} = {} & (q+1, q+3)(q+2, q+4)(q+5, q+8)(q+6, q+7) \\
& (q+9, q+10)(q+11, q+12) \\
h'_{j,3} = {} & (q+1, q+4)(q+2, q+3)(q+5, q+6)(q+7, q+8) \\
& (q+9, q+11)(q+10, q+12)
\end{aligned}
$$

where $q = 2n + 12(k-1)$.

Calculation shows that $W_{l_p}(h'_{j,1}) = W_{l_p}(h'_{j,2}) = W_{l_p}(h'_{j,3}) = \sqrt[p]{6 + 4 \cdot 2^p + 2 \cdot 3^p}$ for $1 \le p < \infty$. Therefore we have the following theorem:

Theorem 13 *The l_1, l_2 and l_p weight problems are* **NP***-complete, even when G is an elementary abelian* 2*-group and each orbit has span at most* 12.

5.3 Lee weight problem

The Lee weight problem is similar to the l_1 weight problem. More precisely, if G is a permutation group on $\{1, \ldots, n\}$ which fixes all points $i > n/2$, then the Lee weight and l_1 weight coincide on G.

Theorem 14 *The Lee weight problem is* **NP***-complete, even when G is an elementary abelian 2-group and each orbit has span at most* 12.

5.4 Kendall's tau and Ulam weight problem

We use the same construction as that for l_p weight problem in Section 5.2. We have $W_I(h'_{j,1}) = W_I(h'_{j,2}) = W_I(h'_{j,3}) = 12$ and $W_U(h'_{j,1}) = W_U(h'_{j,2}) = W_U(h'_{j,3}) = 7$, which imply the following theorem:

Theorem 15 *Kendall's tau and Ulam weight problems are* **NP***-complete, even when G is an elementary abelian* 2*-group.*

6 The l_∞ weight problems

The proof that the l_∞ weight problem is **NP**-complete is similar but a bit more complicated. Part of the reason for this is the following result:

Theorem 16 *The l_∞ maximum weight problem is in* **P**.

PROOF. The l_∞ norm of any permutation in G is bounded above by the maximum span of an orbit of G. Moreover, this bound is attained, since there exists $\pi \in G$ with $\pi(\min(O)) = \max(O)$ for any orbit O. Now the result follows since the orbits can be calculated in polynomial time (they are the connected components of the union of the functional digraphs corresponding to the generators of G).

However, the following holds:

Theorem 17 *The l_∞ weight problem and minimum weight problem for G are both* **NP***-complete, even when G is an elementary abelian group and each orbit has span* 7*.*

PROOF. We use the usual strategy, reduction from **NAESAT**, with an extra trick. The clause gadgets are a bit more complicated. Define permutations h_1, h_2, h_3 by

$$\begin{aligned}
h_1 &= (1,3)(2,4)(5,7)(6,8)(9,13)(10,14)(11,15)(12,16)(17,23)(18,24)\\
&\qquad (19,21)(20,22)\\
h_2 &= (1,5)(2,6)(3,7)(4,8)(9,15)(10,16)(11,13)(12,14)(17,19)(18,20)\\
&\qquad (21,23)(22,24)\\
h_3 &= (1,7)(2,8)(3,5)(4,6)(9,11)(10,12)(13,15)(14,16)(17,21)(18,22)\\
&\qquad (19,23)(20,24)
\end{aligned}$$

Note that each of these has l_∞ weight 6. We also take a permutation

$$\begin{aligned}
g &= (1,8)(2,7)(3,6)(4,5)(9,16)(10,15)(11,14)(12,13)(17,24)(18,23)\\
&\qquad (19,22)(20,21)
\end{aligned}$$

Note that g has weight 7 but gh_i has weight 5 for $i = 1, 2, 3$. We can translate each of these permutations to act on the block $[p+1, p+24]$ in the obvious way without changing the l_∞ weight.

For a variable gadget we can simply take one of the permutations h_i, and the same permutation g.

Now let H be the group produced as in Section 3, and G be generated by H and the element which acts as g on every gadget. Consider the question: does G have an element of norm 5? Such an element must be of the form gh, where h is non-identity on each block; so norm 5 is realised if and only if the group in Section 3 has a FPF element, which we have shown is **NP**-complete.

Since the minimum l_∞-norm of elements of G is either 5 or 6, the **NP**-completeness for minimum norm is also established.

7 Further issues

In this section we consider FPF for transitive groups, other metrics, and an analogue of the Weight Problem for matrix groups.

7.1 Fixed-point-free elements in transitive groups

Although **FPF** is **NP**-complete in arbitrary permutation groups, it is trivial in transitive groups, because of an old result of Jordan [12,18]: for $n > 1$, the answer is always "Yes"; that is, every transitive group of degree $n > 1$ contains a FPF element. It is further known [6] that a proportion at least $1/n$ of the elements of such a group are FPF. We could modify the problem to ask: *How hard is it to find a FPF element*?

There is a very simple randomized algorithm to find a FPF element. If we choose kn elements of G at random, then the probability that we do not find a FPF element is at most

$$\left(1 - \frac{1}{n}\right)^{kn} < \mathrm{e}^{-k},$$

that is, exponentially small.

We conjecture that there is a deterministic polynomial-time algorithm to find a FPF element in a transitive group. The algorithm (based on a proof in [10]) would run as follows:

Step 1: Since blocks of imprimitivity can be found in polynomial time, and since an element of g which fixes no block of imprimitivity must be FPF on points as well, we can reduce to the case where the group G is primitive.

Step 2: Now a minimal normal subgroup N of G is transitive, and is a product of isomorphic simple groups. If N is regular, then any of its elements except e is FPF. Otherwise, one more iteration of Steps 1 and 2 gives a group which is primitive and non-abelian simple.

Step 3: Now we identify the simple group and its action (using the Classification of Finite Simple Groups), and from this knowledge, find a FPF element directly.

For example, suppose that the simple group G is an alternating group A_m, (with its "natural" action on the set $\{1, \dots, m\}$), and let H be the stabiliser of a point in the given action on $\{1, \dots, n\}$. Then H is a maximal subgroup of G. If H contains a 3-cycle (in the natural action), then H is the stabiliser of a subset or a partition of $\{1, \dots, m\}$, and in either case we can choose an element of A_m lying in no conjugate of H. Otherwise, a 3-cycle (in the natural action) is FPF (in the given action).

It seems likely, but is not entirely clear, that Steps 2 and 3 can be done in polynomial time. Certainly the algorithm is by no means simple!

In [10], it is shown that a transitive permutation group of degree greater than 1 contains a FPF element whose order is a power of a prime. The proof of this theorem, unlike Jordan's, requires the Classification of Finite Simple Groups. What is the complexity of finding such an element? The algorithm outlined above may work for this question as well.

We have been unable to decide the complexity of the weight problems for the other metrics considered in this paper restricted to transitive groups.

7.2 Other metrics

If Σ is any set of generators of S_n satisfying $\Sigma = \Sigma^{-1}$, the *Cayley graph* $\mathrm{Cay}(S_n, \Sigma)$ has vertex set S_n and an edge from π to $\sigma\pi$ for any $\pi \in S_n$. The distance function in the Cayley graph is a right-invariant metric. It is left-invariant if and only if Σ is a *normal subset* of S_n, that is, $\pi\Sigma\pi^{-1} = \Sigma$ for all $\pi \in S_n$.

The Cayley metric and Kendall's tau arise in this way, taking Σ to be the set of all transpositions and the set of all adjacent transpositions respectively.

Given a set Σ which can be specified with a polynomial amount of information (for example, Σ of polynomial size or consisting of elements with one of a list of polynomial size of cycle types), we can ask about the Weight Problem for the metric defined by the Cayley graph $\mathrm{Cay}(S_n, \Sigma)$.

Metrics on S_n which are not right-invariant have also been studied. In this case, in place of the weight problem as stated earlier, we ask whether a particular value of the metric is attained as the distance between two elements of the given subgroup G.

For example, the *commutation distance* on S_n is the distance in the *commutation graph*, whose vertex set is $S_n \setminus \{e\}$, with an edge between π and σ if and only if $\pi\sigma = \sigma\pi$. (Since e commutes with every element, extending this metric to all of S_n in the obvious way would make the commutation distance trivial!) Our techniques are of no help in the problem of deciding which values of this distance occur on $G \setminus \{e\}$ for a given group G. This metric is neither right nor left invariant but it is commutation-invariant.

7.3 Complex linear groups

Our questions about Hamming distance for permutation groups can be generalised to linear groups, if we do not require that the "norm function" is derived from a metric. The *character* of a complex linear representation of a group G is the function χ, where $\chi(g)$ is the trace of the matrix representing g, for $g \in G$. Note that any permutation group has a natural matrix representation (by permutation matrices); the character of this representation is given by $\chi(g) = \mathrm{fix}(G)$, the number of fixed points.

So the analogue of the Weight Problem is: Given matrices generating a group G (over the complex numbers) and a complex number c, is there an element $g \in G$ with $\chi(g) = c$? This problem is **NP**-complete since it includes the Hamming weight problem for permutation groups.

There is also an analogue to the material in Section 7.1. A theorem of Burnside [3, p.319] shows that, if the complex representation of G with degree greater than 1 is irreducible, then there is an element $g \in G$ with $\chi(g) = 0$. (This is analogous to Jordan's result, but is not a generalisation since the representation of a transitive permutation group by permutation matrices is not irreducible.) So we can ask the question: what is the complexity of finding such an element?

7.4 Eigenvalue-free matrix

In linear groups over finite fields, eigenvalue-free matrices (that is, matrices having no eigenvalues) play a simialr role to fixed-point-free matrices in permutation groups. See, for example, the enumeration results for classical groups in [13].

So we want to consider the following problem in matrix groups corresponding to the **FPF** problem in permutation groups:

Problem 18 *Eigenvalue-Free (***EF***)*
Instance: Generators for a matrix group M.
Question: Whether M contains an Eigenvalue-free matrix.

Now we have the following theorem:

Theorem 19 *The Eigenvalue-free problem (***EF***) is* **NP***-complete.*

PROOF. We follow the proof for **FPF**, using matrices rather than permutations as our variable and clause gadgets.

A matrix is eigenvalue-free if and only if it acts fixed-point-freely on the projective space. Now the projective line over $\mathbb{F}_3$ contains four points, and admits a Klein four-group, induced by the quaternion subgroup $H = \{\pm I, \pm a_1, \pm a_2, \pm a_3\}$ of $\mathrm{GL}(2, \mathbb{F}_3)$, where

$$a_1 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \quad a_2 = \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \quad a_3 = \begin{pmatrix} 1 & -1 \\ -1 & -1 \end{pmatrix}.$$

We have $a_1^2 = a_2^2 = a_3^2 = a_1 a_2 a_3 = -I$.

Let $\bar{a}$ denote the image of a in $\mathrm{PGL}(2, \mathbb{F}_3)$, and $\bar{H} = \{\bar{a} : a \in H\} = H/\mathbb{F}_3^*$. Then $\bar{H}$ is isomorphic to the Klein group $\mathbb{Z}_2 \times \mathbb{Z}_2$, as is the subgroup formed by the clause gadgets in Section 4.

Now it's easy to check that a_1, a_2 and a_3 are eigenvalue-free. Take one of them, say a_1, as the variable gadget. Similar to the proof in Section 4, we construct a matrix group $M = \langle M_1, M_1', \cdots, M_n, M_n' \rangle$ with each M_t of size $(2n + 2m) \times (2n + 2m)$ for an arbitrary instance (U, C) of **NAESAT** with $|U| = n$ and $|C| = m$. The matrix M_t has the following block structure:

$$M_t = \mathrm{diag}\{A_1, A_2, \cdots, A_n, B_1, \cdots, B_m\},$$

where A_t is the variable gadget a_1 and all other A_i $(i \neq t)$ are I, and B_l is a_j for some $1 \leq j \leq 3$ iff x_t appears in the j-th position of clause c_l and the others are I. The same method is used to construct M_t' while we should notice that B_l is a_j iff $\bar{x}_t$ appears in the j-th position of clause c_l.

The proof that (U, C) has a satisfying assignment for **NAESAT** if and only if M contains an eigenvalue-free matrix is similar to that in Section 4 and we will leave it as an exercise to the reader.

7.5 Other properties

We conclude this section with a general observation. For some cycle structures, deciding whether a permutation group given by a set of generators contains a permutation with such structure is **NP**-complete. For example, in Section 4, we showed that **FPF** is **NP**-complete, and in our example, a FPF element is necessarily a product of 2-cycles. On the other hand, the total number of cycles of a permutation π, including fixed points, is $n - w_T(\pi)$. Therefore from Section 5.1, to decide whether G contains an element with a specified number of cycles is **NP**-complete.

Similarly we can "translate" problems concerning metrics into related properties. For Ulam's metric, we can define an associate sequence s of a permutation π to be a longest increasing subsequence in $(\pi^{-1}(1), \ldots, \pi^{-1}(n))$. Then to decide whether G contains a permutation with associate sequence with given length is hard. The same approach can be used to find hard problems for other metrics.

8 Conclusions and Further Directions

The main contribution of this paper is that we study the (maximal and minimal) weight problems for some well-known metrics on permutation groups. For the maximal weight problem, we showed that l_∞ is in **P** while all other metrics in Section 2 are **NP**-complete. For the (minimal) weight problem, all metrics in Section 2 are **NP**-complete.

For these metrics, there is a dichotomy: each problem is either in **P** or **NP**-complete. Is there a general dichotomy theorem for problems of this type?

The complexity of weight problems also suggests some links between metrics. One interesting case is that the l_p weight (for $1 \leq p < \infty$) is a decreasing function of p, and its limit is the l_∞-weight as $p \to \infty$. The maximum weight problem is hard for l_p-weight but easy for l_∞-weight. One promising direction would be to use the l_∞-weight to approximate the l_p-weight for large p.

One natural consequence of the **NP**-completeness of weight problems is that we know the corresponding counting problems are **#P**-complete. For example, it's easy to show **#FPF** is **#P** complete: our transformation from **NAE-SAT** is parsimonious. (See Welsh [20] for background.) However, when G is transitive, the **FPF** problem is trivial but the complexity of **#FPF** remains unknown though the approximation is easy via a conclusion in [6].

Another consequence is that to decide whether G contains some element with specified cycle structure is often **NP**-complete. (The total number of cycles, including fixed points, of a permutation π is $n - w_T(\pi)$. In an elementary abelian 2-group, any element has cycles of lengths 1 and 2 only, and the numbers of cycles of each length are monotonic functions of the total number.)

Finally, the groups G appearing in all our constructions are abelian. In general, computation in nonabelian groups is harder than that in abelian groups. It would be interesting to understand the role of commutativity in these issues.

Acknowledgements

We are grateful to R.F. Bailey for turning our attention to reference [15], C. Buchheim for conversation about [2], and C.R. Leedham-Green for discussions on Section 7.4. The second author would like to thank S. Riis for help during the preparation of the paper.

References

[1] Berlekamp, E.R., McEliece, R.J. and van Tilborg, H.C.A., *On the inherent intractibility of certain coding problems*, IEEE Trans. Inform. Theory. **IT-24** (1978), 384–386.

[2] Buchheim, C. and Jünger, M., *Linear Optimization over Permutation Groups*, Discrete Optimization, **2** (2005), 308–319.

[3] Burnside, W., “Theory of Groups of Finite Order”, Cambridge University Press, 1911 (reprinted by Dover Publications, New York, 1955).

[4] Buchheim, C., Cameron, P.J., and Wu, T., *On the Subgroup Distance Problem*, in preparation.

[5] Cameron, P.J., “Permutation groups”, Cambridge University Press, 1999.

[6] Cameron, P.J., and Cohen, A.M., *On the number of fixed point free elements in a permutation group*, Discrete Math. **106/107**, (1992), 135–138.

[7] Cook, S., *The complexity of theorem-proving procedures*, Conference Record of Third Annual ACM Symposium on Theory of Computing, 1971, 151–158.

[8] Deza, M. and Huang, T., *Metrics on Permutations: a Survey*, J. Combin. Inform. System Sci. **23** (1998), 173–185.

[9] Diaconis, P., “Group Representations in Probability and Statistics”, Institute of Mathematical Statistics, 1988.

[10] Fein, B., Kantor, W.M. and Schacher, M., *Relative Brauer groups, II*, J. Reine Angew. Math. **328** (1981), 39–57.

[11] Garey, M. and Johnson, D., “Computers and Intractability; A Guide to the Theory of NP-Completeness,” WH Freeman and Company, 1979.

[12] Jordan, C., *Recherches sur les substitutions,* J. Liouville **17**, (1872), 151-367.

[13] Neumann, P.M. and Praeger, C.E., *Derangements and eigenvalue-free elements in finite classical groups*, J. London Math. Soc. (2) **58** (1998), 564–586.

[14] Papadimitriou, C.H., "Computational Complexity", Addison-Wesley Publishing Company, Inc. 1994

[15] Pinch, R.G.E., *The distance of a permutation from a subgroup of S_n*, Combinatorics, Probability and Computing, to appear, 2006.

[16] Praeger, C.E., *On permutation groups with bounded movement*, J. Algebra **144** (1991), 436–442.

[17] Seress, Á., "Permutation group algorithms," Cambridge University Press,2003.

[18] Serre, J.-P., *On a theorem of Jordan*, Bull. Amer. Math. Soc. **40** (2003), 429–440.

[19] Vardy, A., *The intractability of computing the minimum distance of a code*, IEEE Trans. Inform. Theory **43** (1997), 1757–1766.

[20] Welsh, D.J.A., "Complexity : Knots, Colourings and Counting", London Mathematical Society Lecture Note Series **186**, Cambridge University Press, Cambridge, 1993.