The complexity of the Weight Problem for permutation 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 $g \in G$ such that $d(g, e) = k$ for some $k \in \mathbb{N}$. In this paper we show that this problem is **NP**-complete for many well known metrics.

Keywords: Weight Problem, Metrics, Permutation Group, **NP**-complete

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

1 Introduction

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

Problem 1.1 *d-Weight Problem*
Instance: Generators for G in the form of product cycles and $k \in \mathbb{N}$.
Question: Whether there is an element $g \in G$ such that $w_d(g) = k$.

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 many information, such as $|G|$ and the membership test, can be obtained by the Schreier–Sims algorithm in polynomial time [3]. And there are also many other polynomial algorithms obtained for different aspects of the property of G. For further information, [12] is a good resource.

Despite the discovery of these polynomial algorithms, many complexity issues concerned 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 [1]. On the other hand, the complexity of subgroup distance problem was discussed by Pinch for Cayley metric in [11], and later been generalized to other cases by Buchheim etc. in [2].

For the remainder of this paper, we will survey some metrics on permutation group in Section 2. The **NP**-completeness of the Hamming weight problem is proved in Section 3 and of other metrics in Section 4. The maximal version of the weight problem is discussed in Section 5. Finally, in Section 6 we conclude with some open problems.

2 Some metrics on permutation groups

A metric d on S_n is called *right-invariant* if $d(a, b) = d(ac, bc)$ for any $a, b, c \in S_n$. If d is right-invariant, then $d(a, b) = d(ab^{-1}, e) = w_d(ab^{-1})$. In this section we will survey some well known right-invariant metrics on S_n. For more detailed discussion, we recommend [7,6].

- *Hamming Distance*: $H(\pi, \sigma) = |\{i|\pi(i) \neq \sigma(i)\}|$.
- *Cayley Distance*: $T(\pi, \sigma) =$ minimum number of transpositions taking π to σ.
- *Footrule*: $l_1(\pi, \sigma) = \sum_{i=1}^{n} |\pi(i) - \sigma(i)|$.

- $l_\infty(\pi,\sigma) = \max_{1\le i\le n} |\pi(i) - \sigma(i)|$.
- *Spearman's rank correlation*: $l_2(\pi,\sigma) = \sqrt{\sum_{i=1}^{n}(\pi(i) - \sigma(i))^2}$.
- *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*: $U(\pi,\sigma) = n -$ the length of longest increasing subsequence in $(\sigma\pi^{-1}(1), \cdots, \sigma\pi^{-1}(n))$.

3 Hamming 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 3.1 *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 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 [10] defined as:

Problem 3.2 *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 $1 \le i \le m$.
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

- $h_{j,1} = (p+1, p+2)(p+3, p+4)$
- $h_{j,2} = (p+1, p+3)(p+2, p+4)$
- $h_{j,3} = (p+1, p+4)(p+2, p+3)$

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'.

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 3.3 $\mathrm{FPF}(G) \neq \emptyset$ *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

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

where $y_j = 0$ if $t(x_1) = \mathrm{F}$ 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 $g \in \mathrm{FPF}(G)$, then g can be expressed as $g_1^{y_1} {g'_1}^{1-y_1} \cdots g_n^{y_n} {g'_n}^{1-y_n}$ for some $(y_1, \cdots, y_n)$ where $y_k \in \{0, 1\}$ because each generator is of order 2. Then we can show that the assignment t corresponding to $(y_1, \cdots, y_n)$ is a truth assignment satisfying the condition of **NAESAT**. □

The above lemma implies:

Theorem 3.4 **FPF** *is* **NP***-complete.*

Our construction shows more:

Corollary 3.5 **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 Hamming weight problem, now we obtain the following theorem:

Theorem 3.6 *The Hamming weight problem is* **NP***-complete, even when G is an elementary abelian 2-group and each orbit has size at most 4.*

4 The Weight Problem for other metrics

In this section we will consider the weight problems corresponding to the metrics defined in the Section 2.

4.1 Cayley weight problem

Lemma 4.1 *For an elementary abelian 2-group G, we have*

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

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 both Hamming and Cayley weights, while 2-cycles contribute 2 to the Hamming weight and 1 to the Cayley weight. □

Theorem 4.2 *The Cayley weight problem is* **NP***-complete, even when G is an elementary abelian 2-group and each orbit has size at most 4.*

4.2 l_1 and l_2 weight problem

We define the *span* of an orbit to be the size of the interval between its minimum and maximum element.

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:

- $h'_{j,1} = (p+1, p+2)(p+3, p+4)(p+5, p+7)(p+6)(p+8)(p+9, p+12)(p+10, p+11)$
- $h'_{j,2} = (p+1, p+3)(p+2, p+4)(p+5, p+8)(p+6, p+7)(p+9, p+10)(p+11, p+12)$
- $h'_{j,3} = (p+1, p+4)(p+2, p+3)(p+5, p+6)(p+7, p+8)(p+9, p+11)(p+10, p+12)$

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

Calculation shows that $W_{l_1}(h'_{j,1}) = W_{l_1}(h'_{j,2}) = W_{l_1}(h'_{j,3}) = 20$ and $W_{l_2}(h'_{j,1}) = W_{l_2}(h'_{j,2}) = W_{l_2}(h'_{j,3}) = 4\sqrt{42}$. Therefore we have the following theorem:

Theorem 4.3 *l_1 and l_2 weight problem is* **NP***-complete, even when G is an elementary abelian 2-group and each orbit has span at most 12.*

In fact, we can define the l_p metric on permutation group for any $p \in \mathbb{N}$ by $l_p(\pi, \sigma) = \sqrt[p]{\sum_{i=1}^{n} (\pi(i) - \sigma(i))^p}$ and it's easy to show the l_p weight problem is **NP**-complete.

4.3 l_∞ weight problem

The proof that the l_∞ weight problem is **NP**-complete is similar but a bit more complicated and will appear elsewhere. We note one difference: the l_∞ maximum weight problem is in **P**. For the maximum l_∞-weight is just the maximum span of an orbit of G, and the orbits can be computed in polynomial time.

4.4 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 $\alpha > n/2$, then the Lee weight and l_1 weight coincide on G.

Theorem 4.4 *The Lee weight problem is* **NP***-complete, even when G is an elementary abelian 2-group and each orbit has span at most 12.*

4.5 Kendall's tau and Ulam weight problem

We use the same construction as that for l_p weight problem in Section 4.2. 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 4.5 *Kendall's tau and Ulam weight problems are* **NP***-complete, even when G is an elementary abelian 2-group.*

5 Maximal Weight Problem

Let $MaxW_d(G) = \max\{w_d(g)|g \in G\}$. The maximal version of weight problem is defined as follows.

Problem 5.1 *d Maximal Weight Problem*
Instance: Generators for G in the form of product cycles and a $k \in \mathbb{N}$.
Question: Whether the maximal d weight value in G is k.

From the constructions in Section 3 and 4, we know:

Theorem 5.2 *For all metrics in Section 2 except l_∞, the corresponding maximal weight problems are* **NP***-complete. The l_∞ maximal weight problem is in* **P***.*

6 Conclusions and Further Directions

The main contribution of this paper is that we discuss the (maximal) weight problem for most of well known metrics on permutation groups. For maximal weight problem, we show that l_∞ is in **P** while all other metrics in Section 2 are **NP**-complete. For weight problem, all metrics in Section 2 are **NP**-complete except l_∞, which remains open.

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. But when G is transitive, the **FPF** problem is trivial but the complexity **#FPF** remains unknown though the approximation is easy via a conclusion in [4].

Another consequence is that to decide wether G contains some specified structure element turns to be **NP**-complete. For instance, to decide the nontrivial maximal cycles in G is **NP**-complete.

Finally, from our construction, the G is abelian but generally we know that the computation in in nonabelian group is harder than that in abelian case. It

would be interesting to understand the role of commutativity in these issues.

Acknowledgement

We are grateful to R.F. Bailey for turning our attention to reference [11], C. Buchheim for conversation about [1] and S. Riis for fruitful discussions.

References

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

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

[3] Cameron, P.J., "Permutation groups," Cambridge University Press, 1999.

[4] 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.

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

[6] Deza, M. and Huang, T., *Metrics on Permutations*, a Survey.

[7] Diaconis, P., "Group Representations in Probability and Statistics," Institute of Mathematical Statistics, 1988.

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

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

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

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

[12] Seress, A., "Permutation group algorithms," Cambridge University Press,2003.