

# Algorithmic Problems for Metrics on Permutation Groups

V. Arvind and Pushkar S. Joglekar

Institute of Mathematical Sciences  
C.I.T Campus, Chennai 600 113, India  
{arvind, pushkar}@imsc.res.in

**Abstract.** Given a permutation group  $G \leq S_n$  by a generating set, we explore MWP (the minimum weight problem) and SDP (the subgroup distance problem) for some natural metrics on permutations. These problems are known to be NP-hard. We study both exact and approximation versions of these problems. Analogous questions for codes and lattices have been intensively studied in recent years. Since lattices and codes are basically abelian groups, our primary motivation is to see how the techniques applied to lattice problems can be extended to permutation groups (which are nonabelian in general). We summarize our main results:

- For our upper bound results we focus on the Hamming and the  $l_\infty$  permutation metrics. For the  $l_\infty$  metric, we give a randomized  $2^{O(n)}$  time algorithm for finding an optimal solution to MWP. Interestingly, this algorithm adapts ideas from the Ajtai-Kumar-Sivakumar algorithm for the shortest vector problem in lattices [AKS01]. For the Hamming metric, we again give a  $2^{O(n)}$  time algorithm for finding an optimal solution to MWP. This algorithm is based on the classical Schrier-Sims algorithm for finding pointwise stabilizer subgroups of permutation groups.
- It is known that SDP is NP-hard ([BCW06]) and it easily follows that SDP is hard to approximate within a factor of  $\log^{O(1)} n$  unless  $P = NP$ . In contrast, we show that SDP for approximation factor more than  $n/\log n$  is not NP-hard unless there is an unlikely containment of complexity classes.
- For several permutation metrics, we show that the minimum weight problem is polynomial-time reducible to the subgroup distance problem for *solvable* permutation groups.

## 1 Introduction

We investigate the computational complexity of two natural problems for metrics on permutation groups given by generating sets. Given a permutation group  $G = \langle A \rangle \leq S_n$  by a generating set  $A$  of permutations, we are interested in the *minimum weight problem* (denoted MWP) and the *subgroup distance problem* (denoted SDP) for natural permutation metrics. These problems were studied by Cameron et al in [BCW06, CW06] and shown to be NP-hard for several natural permutation metrics.

These problems are analogous to the shortest vector problem and the closest vector problem for integer lattices, and to the minimum Hamming weight problem and nearest codeword problem for linear codes. The corresponding problems for lattices and codes

are also NP-hard, and their approximability is a subject of current intensive study (see e.g. [MG02]). Our primary motivation stems from the fact that lattices and codes are abelian groups, and it is interesting to ask if the upper and lower bound techniques and results for approximability can be extended to arbitrary (nonabelian) permutation groups. Several permutation metrics are in the literature and have been studied from a statistical perspective. Deza [DH98] examines permutation metrics from a coding theory perspective by considering subgroups of  $S_n$  as codes.

*Our main results:* We give  $2^{O(n)}$  time algorithms for the minimum weight problem for both the Hamming and the  $l_\infty$  permutation metrics. Notice that a naive brute-force search algorithm can take  $n!$  steps since  $G \leq S_n$  can have up to  $n!$  elements. In the case of Hamming metric, it turns out that we can design a deterministic  $2^{O(n)}$  time algorithm which is *group theoretic* in nature. The algorithm is based on the classical Schrier-Sims algorithm for finding pointwise stabilizer subgroups of permutation groups (see e.g. [Lu93]). However, the problem for  $l_\infty$  metric does not appear amenable to a group-theoretic approach. Our  $2^{O(n)}$  time randomized algorithm for the problem is more geometric. Interestingly, for this algorithm we are able to adapt ideas from the Ajtai-Kumar-Sivakumar algorithm for the shortest vector problem in lattices [AKS01].

A function  $d : S_n \times S_n \mapsto \mathbb{R}$  is a *metric* on the permutation group  $S_n$  if for all  $\pi, \tau, \psi \in S_n$   $d(\pi, \tau) = d(\tau, \pi) \geq 0$  and  $d(\pi, \tau) = 0$  iff  $\pi = \tau$ . Furthermore, the triangle inequality holds:  $d(\pi, \tau) \leq d(\pi, \psi) + d(\psi, \tau)$ .

Let  $e \in S_n$  denote the identity permutation. For  $\tau \in S_n$ ,  $d(e, \tau)$  is the *norm* of  $\tau$  for metric  $d$ , and is denoted by  $\|\tau\|$ . A *right-invariant* metric  $d$  on  $S_n$  satisfies  $d(\pi, \tau) = d(\pi\psi, \tau\psi)$  for all  $\pi, \tau, \psi \in S_n$ . A *left invariant* metric is similarly defined. For a detailed discussion regarding metrics on  $S_n$  we refer to [DH98]. We recall the definitions of permutation metrics studied in this paper.

**Hamming distance:**  $d(\tau, \pi) = |\{i | \tau(i) \neq \pi(i)\}|$ .

$l_p$  **distance:** for  $p \geq 1$ ,  $d(\tau, \pi) = (\sum_{i=1}^n |\tau(i) - \pi(i)|^p)^{1/p}$ .

$l_\infty$  **distance:**  $d(\tau, \pi) = \max_{1 \leq i \leq n} |\tau(i) - \pi(i)|$ .

**Cayley Distance:**  $d(\tau, \pi) =$  minimum number of transpositions taking  $\tau$  to  $\pi$ .

These metrics are right invariant. The Hamming and the Cayley metric are also left invariant.

For  $S \subseteq S_n$  and  $\tau \in S_n$  let  $d(\tau, S) = \min_{\psi \in S} d(\tau, \psi)$ . For  $\tau \in S_n, r \in \mathbb{R}^+$  let  $B_n(\tau, r, d) = \{\pi \in S_n | d(\pi, \tau) \leq r\}$  be the ball of radius  $r$  centred at  $\tau$  for a metric  $d$ . Analogous to the geometric setting, we define the volume  $\text{Vol}(S)$  of a subset  $S \subseteq S_n$  as its size  $|S|$ . For right invariant metric  $d$  we have, for all  $\tau \in S_n, r \geq 0$ ,  $\text{Vol}(B_n(e, r, d)) = \text{Vol}(B_n(\tau, r, d))$ . Next we define Subgroup Distance Problem and Minimum Weight Problem with respect to a metric  $d$ .

**Definition 1.** [CW06, BCW06]

**Subgroup Distance Problem (SDP):** Input instances are  $(G, \tau, k)$ , where  $G \leq S_n$  is given by a generating set,  $\tau \in S_n$ , and  $k > 0$ . Is  $d(\tau, G) \leq k$ ?

**Minimum Weight Problem (MWP):** Input instances are  $(G, k)$ ,  $G \leq S_n$  given by a generating set and  $k > 0$ . Is there a  $\tau \in G \setminus \{e\}$  with  $\|\tau\| \leq k$ ?

We are also interested in approximate solutions to MWP and SDP. For MWP, given  $\gamma > 1$  the problem is to find a  $\pi \in G, \pi \neq e$  such that  $\|\pi\|$  is bounded by  $\gamma$  times the

optimal value. Likewise for SDP. As usual, we can define promise decision versions of SDP and MWP that capture this notion of approximation.

For any permutation metric  $d$ , the promise problem  $\text{GapSDP}_\gamma$  where  $\gamma$  is a function of  $n$ , is defined as follows: inputs are the SDP inputs  $(G, \tau, k)$ . An instance  $(G, \tau, k)$  is a YES instance if there exist  $\psi \in G$  such that  $d(\psi, \tau) \leq k$ . And  $(G, \tau, k)$  is a NO instance if for all  $\psi \in G$ ,  $d(\psi, \tau) > \gamma k$ . The problem  $\text{GapMWP}_\gamma$  is similarly defined. An algorithm *solves* the promise problem if it decides correctly on the YES and NO instances.

## 2 A $2^{O(n)}$ algorithm for MWP over $l_\infty$ metric

We consider the search version of MWP: given  $G \leq S_n$ , the goal is to find a permutation  $\tau \in G \setminus \{e\}$  with minimum norm with respect to a metric  $d$ . We refer to such a  $\tau \in G$  as a *shortest* permutation in  $G$  w.r.t. the metric  $d$ . First we consider the  $l_\infty$  metric and give a  $2^{O(n)}$  time randomized algorithm for finding a shortest permutation for  $G \leq S_n$  given by generating set. The algorithm uses the framework developed in [AKS01] for the shortest vector problem for integer lattices. Regev's notes [Re] contains a nice exposition.

The basic idea is to first pick  $N$  elements of  $G$  independently and uniformly at random, where  $N$  is  $2^{c \cdot n}$  (where the constant  $c$  will be appropriately chosen). Each of these elements is multiplied by a random permutation of relatively smaller norm to get a new set of  $N$  elements. On this set of permutations a sieving procedure is applied in several rounds. The crucial property of the sieving is that after each stage the remaining permutations have the property that the maximum norm is halved and in the process at most  $2^{c' \cdot n}$  elements are sieved out for a small constant  $c'$ .

Thus, repeated sieving reduces the maximum norm until it becomes a constant multiple of norm of shortest permutation of  $G$ . Then we can argue that for some  $\pi_1, \pi_2$  from the final set of permutations,  $\pi_1 \pi_2^{-1}$  will be a shortest permutation with high probability.

First we prove certain volume bound for  $l_\infty$  metric ball, which is crucially used in the algorithm, next we give a procedure to sample permutations from an  $l_\infty$  metric ball uniformly.

**Lemma 1.** *For  $1 \leq r \leq n - 1$  we have,  $r^n / e^{2n} \leq \text{Vol}(B_n(e, r, l_\infty)) \leq (2r + 1)^n$ . Consequently, for any constant  $\alpha < 1$ ,  $\text{Vol}(B_n(e, r, l_\infty)) / \text{Vol}(B_n(e, \alpha r, l_\infty)) \leq 2^{c_1 \cdot n}$ , where  $c_1 = \log_2(3e^2/\alpha)$ .*

*Proof.* Let  $\tau \in B_n(e, r, l_\infty)$ . So,  $|\tau(i) - i| \leq r$  for all  $i$ . Thus, for each  $i$  there are at most  $2r + 1$  choices for  $\tau(i)$ . This implies  $\text{Vol}(B_n(e, r, l_\infty)) \leq (2r + 1)^n$ . Although better bounds can be shown, this simple bound suffices for the lemma. Now we show the claimed lower bound. Let  $n = kr + t$ ,  $0 \leq t \leq r - 1$ . For  $jr + 1 \leq i \leq (j + 1)r$ ,  $0 \leq j \leq k - 1$ ,  $\tau(i)$  can take any value in  $\{jr + 1, jr + 2, \dots, (j + 1)r\}$ . Hence we have  $\text{Vol}(B_n(e, r, l_\infty)) \geq r!^k t! \geq (r^r / e^r)^k t! \geq r^{n-t} t^t / e^n$ . Using some calculus it is easily seen that the function  $y = r^{n-t} t^t$  is minimum at  $t = r/e$ . Hence  $r^{n-t} t^t / e^n \geq r^n / e^{n+r/e} \geq r^n / e^{2n}$ . This proves the first part of lemma. The second part is immediate. ■

We now explain an almost uniform random sampling procedure from  $B_n(e, r, l_\infty)$ . First, we randomly generate a function  $\tau \in [n]^{[n]}$  by successively assigning values to  $\tau(i)$  for  $i \in [n]$  as follows. For each  $i \in [n]$  we have the list  $L_i = \{j | 1 \leq j \leq n, i - r \leq j \leq i + r\}$  of candidate values for  $\tau(i)$ . Thus we have at most  $(2r + 1)^n$  functions from which we uniformly sample  $\tau$ . Of course,  $\tau$  defined this way need not be a permutation, but if it is a permutation then clearly  $\tau \in B_n(e, r, l_\infty)$ . Our sampling procedure outputs  $\tau$  if it is a permutation and outputs “fail” otherwise. By Lemma 1 the probability that  $\tau$  is a permutation is  $\text{Prob}[\tau \in B_n(e, r, l_\infty)] \geq \frac{r^n}{e^{2n}(2r+1)^n} > \frac{1}{24^n} > 2^{-5n}$ . Thus, if we repeat above procedure sufficiently many times ( say  $2^{10n}$  times ) then the sampling procedure will fail with negligible probability, and when it succeeds it uniformly samples from  $B_n(e, r, l_\infty)$ . In summary we have the following lemma.

**Lemma 2.** *There exists a randomized procedure which runs in time  $2^{O(n)}$  and produces an almost uniform random sample from  $B_n(e, r, l_\infty)$ .*

Now we describe the sieving procedure used in the algorithm. Hereafter we denote  $B_n(\psi, r, l_\infty)$  by  $B_n(\psi, r)$  for simplicity.

**Lemma 3.** *[Sieving Procedure] Let  $r > 0$  and  $\{\tau_1, \tau_2, \tau_3, \dots, \tau_N\} \subseteq B_n(e, r)$  be a subset of permutations. Then in  $N^{O(1)}$  time we can find  $S \subset [N]$  of size atmost  $2^{c_1 n}$  for a constant  $c_1$  such that for each  $i \in [N]$  there is a  $j \in S$  with  $l_\infty(\tau_i, \tau_j) \leq r/2$ .*

*Proof.* We construct  $S$  using a greedy algorithm. Start with  $S = \emptyset$  and run the following step for all elements  $\tau_i, 1 \leq i \leq N$ . At the  $i^{\text{th}}$  step we consider  $\tau_i$ . If  $l_\infty(\tau_i, \tau_j) > r/2$  for all  $j \in S$  include  $i$  in set  $S$  and increment  $i$ . After completion, for all  $i \in [N]$  there is a  $j \in S$  such that  $l_\infty(\tau_i, \tau_j) \leq r/2$ . To argue that  $|S| < 2^{c_1 n}$  for constant  $c_1$  we use the volume bound of Lemma 1. The construction of  $S$  implies for distinct indices  $j, k \in S$  that  $l_\infty(\tau_j, \tau_k) > r/2$ . Hence the metric balls  $B_n(\tau_j, r/4)$  for  $j \in S$  are all pairwise disjoint. The right invariance of  $l_\infty$  metric implies  $\text{Vol}(B_n(\tau_j, r/4)) = \text{Vol}(B_n(e, r/4))$ . As  $\tau_j \in B_n(e, r)$ , by triangle inequality we have  $B_n(\tau_j, r/4) \subseteq B_n(e, r + r/4)$  for  $j \in S$ . Hence  $|S| < \frac{\text{Vol}(B_n(e, 5r/4))}{\text{Vol}(B_n(e, r/4))} \leq 2^{c_1 n}$  by Lemma 1, which also gives the constant  $c_1$ . This completes the proof of the lemma. ■

Now we describe our algorithm to find a shortest permutation in  $G$  using Lemma 3. Let  $t$  denote the norm of a shortest permutation in  $G$ . The following claim gives an easy  $2^{O(n)}$  time algorithm when  $t \geq n/10$ .

**Lemma 4.** *If the norm  $t$  of a shortest permutation in  $G$  is greater than  $n/10$  then in time  $2^{O(n)}$  we can find a shortest permutation in  $G$ .*

*Proof.* Consider  $B_n(\tau, t/2)$  for  $\tau \in G$ . By triangle inequality, all  $B_n(\tau, t/2)$  are disjoint. Also, by Lemma 1 we have  $\text{Vol}(B_n(\tau, t/2)) \geq (t/2)^n \cdot e^{-2n} \geq n^n \beta^{-n}$  for some constant  $\beta > 1$ . Since  $|G| \leq |S_n|/\text{Vol}(B_n(\tau, t/2))$ , it follows that  $|G| \leq \beta^n$ . As we can do a brute-force enumeration of  $G$  in time polynomial in  $|G|$ , we can find a shortest permutation in  $2^{O(n)}$  time. ■

Now we consider the case when  $t < n/10$ . We run the algorithm below for  $1 \leq t < n/10$  (the possible values of  $t$ ) and output a shortest permutation in  $G$  produced by the algorithm.

1. Let  $N = 2^{cn}$ . For  $1 \leq i \leq N$ , pick  $\rho_i$  independently and uniformly at random from  $G$ , and pick  $\tau_i$  almost uniformly at random from  $B_n(e, 2t)$ .
2. Let  $\psi_i = \tau_i \rho_i$ ,  $1 \leq i \leq N$ . Let  $Z = \{(\psi_1, \tau_1), (\psi_2, \tau_2), \dots, (\psi_N, \tau_N)\}$ , and let  $R = \max_i \|\psi_i\|$ .
3. Set  $T = [N]$ .
4. While  $R > 6 * t$  do the following steps:
  - (a) Apply the ‘‘sieving procedure’’ of Lemma 3 to  $\{\psi_i \mid i \in T\}$ . Let set  $S \subseteq T$  be the output of sieving procedure.
  - (b) for all  $i \in S$  remove tuple  $(\psi_i, \tau_i)$  from  $Z$ .
  - (c) for all  $i \notin S$  replace tuple  $(\psi_i, \tau_i) \in Z$  by  $(\psi_i \psi_j^{-1} \tau_j, \tau_i)$ , where  $j \in S$  and  $d(\psi_j, \psi_i) \leq R/2$ .
  - (d) set  $R = R/2 + 2t$ .
  - (e)  $T := T \setminus S$ .
5. For all  $(\varphi_i, \tau_i), (\varphi_j, \tau_j) \in Z$ , let  $\varphi_{i,j} = (\tau_j^{-1} \varphi_j)(\tau_i^{-1} \varphi_i)^{-1}$  (which is in  $G$ ). Output a  $\varphi_{i,j}$  with smallest nonzero norm.

In Step 1 of the algorithm, an almost uniform random sampling procedure from  $l_\infty$  metric ball is given by Lemma 2. For  $G \leq S_n$ , uniform sampling from  $G$  can be done in polynomial time by using a strong generating set for  $G$  (see e.g. [Lu93]). A random element is obtained by picking a coset representative at each level from the pointwise stabilizer tower of subgroups and multiplying them out. Thus Step 1 of the algorithm takes  $2^{O(n)}$  time. Clearly, the while loop takes  $2^{O(n)}$  time.

In order to prove the correctness, we examine the invariant maintained during each iteration of the while loop.

**Proposition 1.** *Before each iteration of the while loop, the following invariant is maintained. For all  $i \in T$  we have  $(\varphi_i, \tau_i) \in Z$ ,  $\tau_i^{-1} \varphi_i \in G$  and  $\|\varphi_i\| \leq R$ .*

*Proof.* Clearly, the invariant holds before the first iteration. Inductively, suppose that at the beginning of an arbitrary iteration the set  $Z$  is of the form  $Z = \{(\varphi_i, \tau_i) \mid i \in T\}$  such that  $\tau_i^{-1} \varphi_i \in G$  and  $\|\varphi_i\| \leq R$ . During this iteration, in  $Z$  we replace  $(\varphi_i, \tau_i)$  by  $(\varphi_i \varphi_j^{-1} \tau_j, \tau_i)$ , where  $j \in S$  and  $l_\infty(\varphi_i, \varphi_j) \leq R/2$ . By right invariance of the  $l_\infty$  metric, we have  $l_\infty(\varphi_i, \varphi_j) = \|\varphi_i \varphi_j^{-1}\| \leq R/2$ . Triangle inequality implies  $\|\varphi_i \varphi_j^{-1} \tau_j\| \leq \|\tau_j^{-1}\| + \|\varphi_i \varphi_j^{-1}\| = \|\tau_j\| + \|\varphi_i \varphi_j^{-1}\| \leq 2t + R/2$  which equals the value of  $R$  set in Step 4(d). Hence,  $\|\varphi_i\| \leq R$  at the beginning of next iteration. Clearly,  $\tau_i^{-1} \varphi_i \varphi_j^{-1} \tau_j$  is in  $G$  since  $\tau_i^{-1} \varphi_i$  and  $\tau_j^{-1} \varphi_j$  are in  $G$ . ■

By Proposition 1 when the algorithm stops (after Step 5) we have  $\tau_i^{-1} \varphi_i \in G$  and  $\|\tau_i^{-1} \varphi_i\| \leq 8t$  for all  $(\varphi_i, \tau_i) \in Z$ . We want to argue that one of the  $\varphi_{i,j}$  is equal to a shortest permutation in  $G$  with high probability. In Step 1 we pick  $\tau_i$  almost uniformly at random from  $B_n(e, 2t)$ . As in the Regev’s analysis of AKS algorithm in [Re], we define a new random procedure which also uniformly samples from  $B_n(e, 2t)$  and has some properties which enable us to conveniently argue the correctness of the algorithm. In the lattice setting, the euclidean metric makes it easier to define a modified sampling from  $B_n(e, 2t)$ . However, for the  $l_\infty$  metric over  $S_n$ , the modified sampling from  $B_n(e, 2t)$  is more involved.

Let  $\tau \in G$  be an element with shortest nonzero norm  $t$ . We introduce some notation. Let  $C_\tau = B_n(e, 2t) \cap B_n(\tau, 2t)$ ,  $C_{\tau^{-1}} = B_n(e, 2t) \cap B_n(\tau^{-1}, 2t)$  and  $C = C_\tau \cap C_{\tau^{-1}}$ . The following claim is obvious.

**Lemma 5.** *Consider a map  $\phi_1 : C_\tau \rightarrow C_{\tau^{-1}}$  defined as  $\phi_1(\sigma) = \sigma\tau^{-1}$ . Then  $\phi_1$  is a bijection from  $C_\tau$  onto  $C_{\tau^{-1}}$ .*

Let  $\phi'_1 : C_{\tau^{-1}} \rightarrow C_\tau$  denote the inverse of  $\phi_1$ .

We now define a randomized procedure `Sample` which on input a random permutation  $\sigma \in B_n(e, 2t)$  returns a new random permutation  $\text{Sample}(\sigma) \in B_n(e, 2t)$ .

- (i) If  $\sigma \notin C_\tau \cup C_{\tau^{-1}}$  then  $\text{Sample}(\sigma) = \sigma$  with probability 1.
- (ii) If  $\sigma \in C_\tau \setminus C$  then
  - (a) if  $\phi_1(\sigma) \in C$  then randomly set  $\text{Sample}(\sigma)$  to either  $\sigma$  with probability  $3/4$  or to  $\phi_1(\sigma)$  with probability  $1/4$ .
  - (b) if  $\phi_1(\sigma) \notin C$  then randomly set  $\text{Sample}(\sigma)$  to  $\sigma$  or  $\phi_1(\sigma)$  with probability  $1/2$  each.
- (iii) If  $\sigma \in C_{\tau^{-1}} \setminus C$  then define  $\text{Sample}(\sigma)$  analogously as in Step (ii) above, using  $\phi'_1$  instead of  $\phi_1$ .
- (iv) If  $\sigma \in C$ , then randomly set  $\text{Sample}(\sigma)$  to either  $\sigma$  with probability  $1/2$ , or to  $\phi_1(\sigma)$  with probability  $1/4$ , or to  $\phi'_1(\sigma)$  with probability  $1/4$ .

The following lemma essentially states that the random variables  $\text{Sample}(\sigma)$  and  $\sigma$  are identically distributed.

**Lemma 6.** *If  $\sigma$  is uniformly distributed in  $B_n(e, 2t)$  then  $\text{Sample}(\sigma)$  is also uniformly distributed in  $B_n(e, 2t)$ .*

*Proof.* Let  $V = \text{Vol}(B_n(e, 2t))$ . We claim  $\text{Sample}(\sigma)$  is uniformly distributed over  $B_n(e, 2t)$ . For each  $\pi \in B_n(e, 2t)$  we have

$$\text{Prob}[\text{Sample}(\sigma) = \pi] = \sum_{\delta \in B_n(e, 2t)} \text{Prob}[\sigma = \delta] \cdot \text{Prob}[\text{Sample}(\delta) = \pi].$$

We need to show that  $\sum_{\delta \in B_n(e, 2t)} \text{Prob}[\sigma = \delta] \cdot \text{Prob}[\text{Sample}(\delta) = \pi] = 1/V$ . As  $\sigma$  is uniformly distributed, it is equivalent to showing  $\sum_{\delta \in B_n(e, 2t)} \text{Prob}[\text{Sample}(\delta) = \pi] = 1$ . If  $\pi \notin C_\tau \cup C_{\tau^{-1}}$  it is true directly from the definition of `Sample`. Consider  $\pi \in C$ , since maps  $\phi_1$  and  $\phi'_1$  are bijective, there are unique  $\sigma_1 \neq \sigma_2$  such that  $\phi_1(\sigma_1) = \phi'_1(\sigma_2) = \pi$ . By definition of `Sample` we have  $\text{Prob}[\text{Sample}(\sigma_1) = \pi] = \text{Prob}[\text{Sample}(\sigma_2) = \pi] = \frac{1}{4}$  and  $\text{Prob}[\text{Sample}(\pi) = \pi] = \frac{1}{2}$ . Summing up we get  $\sum_{\delta \in B_n(e, 2t)} \text{Prob}[\text{Sample}(\delta) = \pi] = 1$  as desired. Now suppose  $\pi \in C_\tau \setminus C$ . If  $\phi_1(\pi) = \psi \in C$  then clearly  $\phi'_1(\psi) = \pi$ . The definition of `Sample` implies that  $\sum_{\delta \in B_n(e, 2t)} \text{Prob}[\text{Sample}(\delta) = \pi] = \text{Prob}[\text{Sample}(\psi) = \pi] + \text{Prob}[\text{Sample}(\pi) = \pi] = \frac{1}{4} + \frac{3}{4} = 1$ . If  $\phi_1(\pi) = \psi \notin C$ , we have  $\sum_{\delta \in B_n(e, 2t)} \text{Prob}[\text{Sample}(\delta) = \pi] = \frac{1}{2} + \frac{1}{2} = 1$ . The case when  $\pi \in C_{\tau^{-1}} \setminus C$  is similar. This proves the lemma.  $\blacksquare$

It follows from the definition of `Sample` that replacing  $\tau_i$  by  $\text{Sample}(\tau_i)$  does not affect the distribution of  $\psi_i$  in Step 2. In fact,  $\text{Sample}(\tau_i)$  and  $\tau_i$  are identically distributed

by Lemma 6. In Step 1 we pick each  $\tau_i$  almost uniformly at random from  $B_n(e, 2t)$ . Now, in our analysis we replace this by  $\text{Sample}(\tau_i)$ . The crucial point of the argument is that it suffices to replace  $\tau_i$  by  $\text{Sample}(\tau_i)$  after Step 2, as the  $\tau_i$  is only used to define  $\psi_i$  and it will not affect the distribution of  $\psi_i$  if we replace  $\tau_i$  by  $\text{Sample}(\tau_i)$ . Note that  $\tau_i$  is used during sieving step in the while loop only if  $i$  lies in the sieved set  $S$ . The remaining  $\tau_i$  are replaced by  $\text{Sample}(\tau_i)$  in Step 5. Clearly, this modification does not change the probability of computing a shortest permutation as the distributions in the two cases are the same. Note that  $\text{Sample}(\tau_i)$  is introduced for analysis. We cannot implement the procedure  $\text{Sample}$  efficiently as we do not know  $\tau$ .

In Step 1 of the algorithm we pick each  $\tau_i$  almost uniformly at random from  $B_n(e, 2t)$ . The initial set is  $\{\tau_i \mid i \in [N]\}$ . The while loop iterates for at most  $2 \log n$  steps and in each step we remove a set  $S$  of size at most  $2^{c_1 n}$ , where  $c_1$  is given by Lemma 1. Thus, at the end of the while loop we still have  $N - 2 \log n \cdot 2^{c_1 n}$  many  $\tau_i$  in the remaining set  $T$ . Thus, as argued earlier for the purpose of analysis we can replace  $\tau_i$  by  $\text{Sample}(\tau_i)$  for all  $i \in T$  and it still doesn't affect the working of the algorithm.

The triangle inequality implies  $B_n(e, t) \subseteq C_\tau$ . Thus  $\text{Vol}(C_\tau) \geq \text{Vol}(B_n(e, t)) \geq t^n \cdot e^{-2n}$  by Lemma 1. Also,  $\text{Vol}(B_n(e, 2t)) \leq (5t)^n$ . Hence,  $\frac{\text{Vol}(C_\tau \cup C_{\tau^{-1}})}{\text{Vol}(B_n(e, 2t))} \geq 2^{-c_2 n}$ , for some constant  $c_2$  (which depends on  $c_1$ ). Thus a random  $\pi \in B_n(e, 2t)$  lies in  $C_\tau \cup C_{\tau^{-1}}$  with probability at least  $2^{-c_2 n}$ .

Given a constant  $c_3 > 0$ , we can choose a suitably large  $N = 2^{cn}$  for a constant  $c$  so that at least  $2^{c_3 n}$  many  $\tau_i$  for  $i \in T$  at the end of the while loop will lie in  $C_\tau \cup C_{\tau^{-1}}$ . Thus, with probability  $1 - 2^{-O(n)}$  we can guarantee that at least  $2^{c_3 n}$  many  $\tau_i$  for  $i \in T$  are such that  $\tau_i \in C_\tau \cup C_{\tau^{-1}}$  and  $(\varphi_i, \tau_i) \in Z$  at the beginning of Step 5.

Furthermore, at the beginning of Step 5 each  $(\varphi_i, \tau_i) \in Z$  satisfies  $\|\tau_i^{-1} \varphi_i\| \leq 8t$  and  $\tau_i^{-1} \varphi_i \in G$ . Now we argue using the pigeon-hole principle that there is some  $\pi \in G$  such that  $\pi = \tau_i^{-1} \varphi_i$ ,  $(\varphi_i, \tau_i) \in Z$  for at least  $2^n$  indices  $i \in T$ .

*Claim.*  $|G \cap B_n(e, 8t)| < 2^{c_4 n}$  for some constant  $c_4$ .

*Proof.* Note that  $l_\infty(\pi_1, \pi_2) \geq t$  for distinct  $\pi_1, \pi_2 \in G$ . Thus, metric balls of radius  $t/2$  around each element in  $G \cap B_n(e, 8t)$  are all pairwise disjoint. By triangle inequality, all these  $t/2$  radius metric balls are contained in  $B_n(e, 8t + t/2)$ . Hence  $|G \cap B_n(e, 8t)| < \text{Vol}(B_n(e, 17t/2)) / \text{Vol}(B_n(e, t/2)) < 2^{c_4 n}$ , by Lemma 1. This proves the claim. ■

Let  $c_3 = c_4 + 1$ . Then with probability  $1 - 2^{-O(n)}$  we have  $\pi \in G$  such that  $\pi = \tau_i^{-1} \varphi_i$ ,  $(\varphi_i, \tau_i) \in Z$  for at least  $2^{c_3 n} / 2^{c_4 n} = 2^n$  indices  $i \in T$ . Call this set of indices  $T_0$ .

Recall that in our analysis we can replace  $\tau_i$  by  $\text{Sample}(\tau_i)$  for each  $i \in T_0$ . By the definition of  $\text{Sample}(\tau_i)$ ,  $\text{Prob}[\text{Sample}(\tau_i) = \tau_i \forall i \in T_0] \leq (3/4)^{2^n}$ . Similarly,  $\text{Prob}[\text{Sample}(\tau_i) \neq \tau_i \forall i \in T_0] \leq (1/2)^{2^n}$ . Hence with probability  $1 - 2^{-O(n)}$  there are indices  $i, j \in T_0$  such that  $(\varphi_i, \tau_i), (\varphi_j, \tau_j) \in Z$  and  $\text{Sample}(\tau_i) = \tau_i$   $\text{Sample}(\tau_j) \neq \tau_j$ . Clearly,  $\text{Sample}(\tau_j) = \tau_j \tau$  or  $\text{Sample}(\tau_j) = \tau_j \tau^{-1}$ . Without loss of generality, assume  $\text{Sample}(\tau_j) = \tau_j \tau$ . Then, after Step 5 we have with high probability  $\varphi_{i,j} = ((\tau_j \tau)^{-1} \varphi_j) (\tau_i^{-1} \varphi_i)^{-1} = \tau^{-1} \pi \pi^{-1} = \tau^{-1}$ . In other words, with probability  $1 - 2^{-O(n)}$  one of the  $2^{O(n)}$  output permutations is a ‘‘shortest’’ permutation in  $G$ . We have shown the following theorem.

**Theorem 1.** *Given a permutation group  $G \leq S_n$  as input, we can find a permutation in  $G \setminus \{e\}$  with smallest possible norm with respect to  $l_\infty$  metric in  $2^{O(n)}$  randomized time.*

### 3 Weight Problems for Hamming metric

We first give an easy  $2^{O(n)}$  time deterministic algorithm for Minimum Weight Problem in the case of Hamming metric. It turns out we can use a well-known algorithm from permutation groups. Suppose  $G \leq S_n$  is given by a generator set. The problem is to find a shortest permutation in  $G$  for the Hamming metric. For every  $S \subseteq [n]$  consider the pointwise stabilizer subgroup  $G_S \leq G$  defined as  $G_S = \{g \in G \mid \forall i \in S : g(i) = i\}$ . Using the Schreier-Sims algorithm in polynomial time [Lu93] we can compute a generating set for  $G_S$ . Thus, in  $2^{O(n)}$  time we can compute  $G_S$  for all  $S \subseteq [n]$  and find the largest  $t < n$  for which there is  $S \subseteq [n]$  such that  $|S| = t$  and  $G_S$  is a nontrivial subgroup. Clearly, any  $\tau \neq e \in G_S$  is a shortest permutation with respect to Hamming metric.

Finally, we also consider the problem of finding an element in  $G \leq S_n$  of maximum norm w.r.t. Hamming metric. We first consider the problem of deciding if  $G \leq S_n$  has a fixpoint free permutation. In general it is known that this problem is NP-complete [CW06]. Using Inclusion-Exclusion Principle we give a  $2^{O(n)}$  time deterministic algorithm for the search version of the problem as follows. As before, let  $G_S$  be the subgroup of  $G$  that pointwise fixes  $S \subseteq [n]$ . Let  $F \subseteq G$  denote the set of fixpoint free elements. Clearly,  $F \cap G_S = \emptyset$  for each nonempty  $S$ . Also,  $F \cup \bigcup_{S \neq \emptyset} G_S = G$ . In  $2^{O(n)}$  time we can compute generating sets for all  $G_S$ .

For the algorithm, inductively assume that we have already computed a coset  $H_{k-1}$  of  $G_{[k-1]}$  in  $G$ , where for all  $\tau \in H_{k-1}$  we have  $\tau(i) = \alpha_i$ ,  $\alpha_i \in [n]$  for  $1 \leq i \leq k-1$  and  $H_{k-1}$  contains a fixpoint free permutation in  $G$ .

We now show how to compute a point  $\alpha_k \in [n]$  which will fix the coset  $H_k$  of  $G_{[k]}$  in  $2^{O(n)}$  time such that for all  $\tau \in H_k$ ,  $\tau(i) = \alpha_i$  for  $i = 1$  to  $k$  and  $H_k$  contains a fixpoint free element in  $G$ . By repeating this successively we can find a fixpoint free permutation. First, from the orbit of  $k$  we pick a candidate point  $\alpha_k$  distinct from  $\alpha_1, \dots, \alpha_{k-1}$  and  $k$ . Let  $H_k = \{\tau \in G \mid \tau(i) = \alpha_i, 1 \leq i \leq k\}$ .

Let  $A_i = H_k \cap G_{\{i\}}$  for  $i = k+1$  to  $n$ . It is clear that intersection of any subcollection of these  $A_i$ 's is of the form  $H_k \cap G_S$  for  $S \subseteq [n]$ . We can compute  $G_S$  in polynomial time for any  $S \subseteq [n]$  using the Schreier-Sims algorithm [Lu93]. Furthermore, the coset intersection problem  $H_k \cap G_S$  can also be solved in  $2^{O(n)}$  time using the machinery of Babai and Luks [BL83, Lu93]. Thus, in time  $2^{O(n)}$  we can compute  $|\bigcap_{i \in S} A_i|$  for all subsets  $S \subseteq [n-k]$ . In  $2^{O(n)}$  further steps, by using the Inclusion-Exclusion formula, we can compute  $|A_{k+1} \cup A_{k+2} \cup \dots \cup A_n| = m$ . Clearly,  $H_k$  contains a fixpoint free element of  $G$  iff  $m < |H_k|$ . If  $m = |H_k|$ , we try the next candidate value for  $\alpha_k$  from the orbit of  $k$ . This procedure clearly succeeds assuming  $H_{k-1}$  has a fixpoint free element of  $G$ . This gives  $2^{O(n)}$  time algorithm to find a fixpoint free permutation. With minor changes to this algorithm, we can compute an element of maximum norm in  $G$  with respect to Hamming norm in  $2^{O(n)}$  time. We summarize these observations in the following theorem.

**Theorem 2.** *Given a permutation group  $G \leq S_n$  by a generating set, in  $2^{O(n)}$  time we can find  $\tau \in G \setminus \{e\}$  with smallest possible norm and  $\psi \in G$  with largest possible norm with respect to Hamming metric.*

## 4 MWP is reducible to SDP for solvable permutation groups

For integer lattices, SVP (shortest vector problem) is polynomial-time reducible to CVP (closest vector problem) [GMSS99]. A similar result for linear codes is also proved there. We show an analogous result for *solvable* permutation groups. In fact we give a polynomial-time Turing reduction from MWP to SDP, which works for the gap version of the problem for *any* right invariant metric  $d$ . We do not know if this reduction can be extended to nonsolvable permutation groups. Finally we make an observation about the hardness of approximation of SDP and MWP.

Let  $G \leq S_n$  be input instance for MWP. The idea is to make different queries of the form  $(H, \tau)$  to SDP, for suitable subgroups  $H \leq G$  and  $\tau \notin H$ .

Let  $d$  be a right invariant metric on  $S_n$ . We want to find a shortest permutation  $\tau \in G$  w.r.t. metric  $d$ . It is well-known in algorithmic permutation group theory (e.g. see [Lu93]) that for solvable permutation groups  $G \leq S_n$  we can compute in deterministic polynomial time a composition series  $G = G_k \triangleright G_{k-1} \triangleright \dots \triangleright G_1 \triangleright G_0 = \{e\}$ ,  $k \leq n$ . In other words,  $G_{i-1}$  is a normal subgroup of  $G_i$  for each  $i$ . Furthermore, since  $G$  is solvable, each quotient group  $G_i/G_{i-1}$  has prime order, say  $p_i$  (where the  $p_i$ 's need not be distinct). Notice that for any  $\tau_i \in G_i \setminus G_{i-1}$ , the coset  $G_{i-1}\tau_i$  generates the cyclic quotient group  $G_i/G_{i-1}$ . It is easily seen that these elements  $\tau_i$  form a generating set for  $G$  with the following standard property. We omit proof due to lack of space.

**Proposition 2.** *For each  $i$ ,  $1 \leq i \leq k$ , every  $\tau \in G_i \setminus G_{i-1}$  can be uniquely expressed as  $\tau = \tau_1^{\alpha_1} \tau_2^{\alpha_2} \dots \tau_i^{\alpha_i}$ ,  $0 \leq \alpha_j < p_j$ ,  $1 \leq j \leq i$  and  $\alpha_i \neq 0$ .*

**Theorem 3.** *For any right invariant metric  $d$  on  $S_n$ , and for solvable groups, GapMWP $_\gamma$  is polynomial time Turing reducible to GapSDP $_\gamma$ .*

*Proof.* Let  $(G, m)$  be an input instance of GapMWP $_\gamma$ . We compute  $\tau_1, \dots, \tau_k$  for the group  $G$  as described above. Then we query the oracle of GapSDP $_\gamma$  for instances  $(G_{i-1}, \tau_i^{-r}, m)$ , for  $1 \leq i \leq k$ ,  $1 \leq r < p_i$ . The reduction outputs “YES” if at least one of the queries answers “YES” otherwise it outputs “NO”. Clearly, the reduction makes at most  $O(n^2)$  oracle queries and runs in polynomial time. We prove its correctness.

Suppose  $(G, m)$  is a “YES” instance of GapMWP $_\gamma$ . We show that at least one of the queries  $(G_{i-1}, \tau_i^{-r}, m)$ ,  $1 \leq i \leq k$ ,  $1 \leq r < p_i$  will return “YES”. Let  $\tau \in G = G_k$  such that  $\|\tau\| \leq m$ . Let  $i$  be the smallest such that  $\tau \notin G_{i-1}$ ,  $\tau \in G_i$ . From Proposition 2 it follows that  $\tau$  can be uniquely expressed as  $\prod_{j=1}^i \tau_j^{\alpha_j}$ , where  $0 \leq \alpha_j < p_j$ ,  $1 \leq j \leq i$  and  $\alpha_i \neq 0$ . As  $\prod_{j=1}^{i-1} \tau_j^{\alpha_j} \in G_{i-1}$ , we have  $d(\tau_i^{-\alpha_i}, G_{i-1}) \leq d(\tau_i^{-\alpha_i}, \prod_{j=1}^{i-1} \tau_j^{\alpha_j})$ . The right invariance of  $d$  implies  $d(\tau_i^{-\alpha_i}, G_{i-1}) \leq d(e, \prod_{j=1}^i \tau_j^{\alpha_j}) = \|\tau\| \leq m$ . Hence  $(G_{i-1}, \tau_i^{-\alpha_i}, m)$  is a “YES” instance of GapSDP $_\gamma$ .

Now suppose  $(G_{i-1}, \tau_i^{-r}, m)$ ,  $1 \leq i \leq k, 1 \leq r \leq p_i - 1$  is not a “NO” instance of  $\text{GapSDP}_\gamma$ . Then there is some  $\tau \in G_{i-1}$  such that  $d(\tau, \tau_i^{-r}) \leq \gamma m$ , i.e.  $\|\tau \tau_i^r\| \leq \gamma m$ . As  $\tau_i \in G_i \setminus G_{i-1}$ ,  $\tau_i^t \notin G_{i-1}$  for  $1 \leq t \leq p_i - 1$ . Thus  $\tau_i^r \notin G_{i-1}$  implying  $\tau \tau_i^r \neq e$ . Hence  $(G, m)$  is not a “NO” instance of  $\text{GapMWP}_\gamma$ . This completes the proof. ■

Cameron et al [BCW06,CW06] have shown that SDP and MWP are NP-hard for several permutation metrics. It follows from [ABSS97] that SDP for linear codes is NP-hard to approximate within a factor of  $(\log n)^c$ , where  $n$  is the block length of the input code and  $c$  is an arbitrary constant. Furthermore, Dumer et al [DMS99] have shown that constant-factor approximation is NP-hard for MWP restricted to binary linear codes. Given a binary linear code  $C$  of block length  $n$ , we can easily construct an abelian 2-group  $G \leq S_{2n}$  isomorphic to  $C$ . An easy consequence of this construction and known hardness results for binary linear codes directly yields the following hardness results for  $\text{GapSDP}_\gamma$  and  $\text{GapMWP}_\gamma$  for different metrics.

**Theorem 4.** *For Hamming, Cayley, and the  $l_p$  metrics,  $\text{GapSDP}_\gamma$  is NP-hard for  $\gamma = O((\log n)^c)$  and  $\text{GapMWP}_\gamma$  is NP-hard under randomized reduction for any constant  $\gamma$ .*

## 5 Limits of hardness

Since  $\text{GapSDP}_\gamma$  is NP-hard for  $\gamma \leq (\log n)^c$ , a natural question is to explore its complexity for larger gaps. For the  $\text{GapCVP}$  problem on lattices, Goldreich and Goldwasser [GG00] have shown a constant round IP protocol for  $O(\sqrt{n/\log n})$  gap in the case of  $l_2$  norm. Consequently, for this gap  $\text{GapCVP}$  is not NP-hard unless polynomial hierarchy collapses. We adapt similar ideas to the permutation group setting. For the Hamming and Cayley metric we give a constant round IP protocol for the complement problem of  $\text{GapSDP}_\gamma$  for  $\gamma \geq n/\log n$ , such that the protocol rejects “YES” instances of  $\text{GapSDP}_\gamma$  with probability at least  $n^{-\log n}$ , and always accepts the “NO” instances. For designing the IP protocols we require uniform random sampling procedures from metric balls for the Hamming and Cayley metrics.

We first consider the Cayley metric. Recall that the Cayley distance between  $\tau$  and  $e$  is the least number of transpositions required to take  $\tau$  to  $e$ . Let  $k$  be the number of cycles in  $\tau$ . Each transposition multiplied to  $\tau$  increases or decreases the number of cycles by 1. Since  $\tau$  is transformed to  $e$  with the fewest transpositions if we always multiply by a transposition that increments the number of cycles, we have  $d(e, \tau) = n - k$ . Thus, a Cayley metric ball of radius  $r$  contains  $\tau \in S_n$  such that  $\tau$  has at least  $n - r$  cycles. The number  $c(n, k)$  of permutation in  $S_n$  with exactly  $k$  cycles is a Stirling number of the first kind and it satisfies the recurrence relation  $c(n, k) = (n - 1)c(n - 1, k) + c(n - 1, k - 1)$ . We can compute  $c(m, l)$ ,  $0 \leq m \leq n, 0 \leq l \leq k$  using the recurrence for  $c(n, k)$ .

**Proposition 3.** *Let  $S \subseteq S_n$  be the set of permutations with  $k$  cycles. Let  $N = |S| = c(n, k)$ . Then there exists a polynomial (in  $n$ ) time computable bijective function  $f_{n,k} : [N] \mapsto S$ .*

*Proof.* If  $n = k = 1$ , clearly such function exists,  $f_{1,1}(1)$  is simply defined as identity element of  $S_1$ . We use induction on  $n+k$ . Assume that such functions exist for  $n+k \leq t$ . Now consider  $n, k$  such that  $n+k = t+1$ . We define the function  $f_{n,k}(i)$ , for  $1 \leq i \leq N$ :

1. If  $i > (n-1)c(n-1, k)$ , let  $\pi = f_{n-1, k-1}(i - (n-1)c(n-1, k))$  and  $\tau$  be obtained by appending a 1-cycle  $(n)$  to  $\pi$ . Define  $f_{n,k}(i) = \tau$ .
2. If  $i \leq (n-1)c(n-1, k)$  then find  $j$  such that  $(j-1)c(n-1, k) < i \leq jc(n-1, k)$ . Let  $\pi = f_{n-1, k}(i - (j-1)c(n-1, k))$ , write  $\pi$  as product of disjoint cycles. Let  $\tau \in S_n$  be obtained by inserting  $n$  in the  $j^{\text{th}}$  position of the cyclic decomposition of  $\pi$ . Define  $f_{n,k}(i) = \tau$ .

Clearly,  $f_{n,k}$  is polynomial time computable. We show  $f_{n,k}$  is bijective by induction. Suppose  $f_{n-1, k-1}$  and  $f_{n-1, k}$  are bijective. Each  $\tau \in S_n$  with  $k$  cycles can be uniquely obtained either by inserting element  $n$  in cyclic decomposition of a  $\pi \in S_{n-1}$  with  $k$  cycles (which can be done in  $n-1$  ways) or by attaching a 1-cycle with element  $n$  to some  $\pi \in S_{n-1}$  with  $k-1$  cycles. It follows that  $f_{n,k}$  is bijective. ■

To uniformly sample  $\tau \in S_n$  with  $k$  cycles, we pick  $m \in \{1, 2, \dots, c(n, k)\}$  uniformly at random and let  $\tau = f_{n,k}(m)$ .

**Lemma 7.** *There is a randomized procedure which runs in time  $\text{poly}(n)$  and samples from  $B_n(e, r, d)$  almost uniformly, where  $d$  denotes Cayley metric.*

Now consider the Hamming Metric. The Hamming ball of radius  $r$  contains all  $\tau \in S_n$  such that  $\tau(i) \neq i$  for at most  $r$  many points  $i$ . Hence,  $\text{Vol}(B_n(e, r, d)) = \sum_{i=0}^r \binom{n}{i} D_i$ , where  $D_i$  denotes the number of derangements on  $i$  points. We can easily enumerate all  $i$ -element subsets of  $[n]$ . The number  $D_i$  of derangements on  $i$  points satisfies the recurrence  $D_i = (i-1)(D_{i-1} + D_{i-2})$ . With similar ideas as used for sampling for Cayley metric balls we can do almost uniform random sampling from Hamming metric balls in polynomial time.

**Lemma 8.** *For  $r > 0$ , there exists a randomized procedure which runs in time  $\text{poly}(n)$  and samples almost uniformly at random from the Hamming balls of radius  $r$  around  $e$  ( $B_n(e, r, d)$ ).*

We now describe the simple 2-round IP protocol for the Hamming metric. Let  $(G, \tau, r)$  be input instance of  $\text{GapSDP}_\gamma$  for  $\gamma \geq n/\log n$ , and  $d$  is the Hamming metric.

1. **Verifier:** picks  $\sigma \in \{0, 1\}$ ,  $\psi \in G$ ,  $\beta \in B_n(e, \gamma r/2, d)$  almost uniformly at random. The verifier sends to the prover the permutation  $\pi = \beta\psi$  if  $\sigma = 0$ , and  $\pi = \beta\tau\psi$  if  $\sigma = 1$ .
2. **Prover:** The prover sends  $b = 0$  if  $d(\pi, G) < d(\pi, \tau G)$  and  $b = 1$  otherwise.
3. **Verifier:** Accepts iff  $b = \sigma$ .

For the protocol we need polynomial time random sampling from a permutation group which is quite standard [Lu93]. We also need uniform sampling from Hamming metric balls which is given by Lemma 8.

We omit the proof of correctness of the next lemma due to lack of space.

**Lemma 9.** *The verifier always accepts if  $(G, \tau, r)$  is “NO” instance of  $\text{GapSDP}_\gamma$ . Furthermore, the verifier rejects with probability at least  $n^{-\log n}$  if  $(G, \tau, r)$  is a “YES” instance of  $\text{GapSDP}_\gamma$ .*

This shows correctness of the protocol for Hamming metric. For the Cayley metric too a similar IP protocol can be designed. As an immediate consequence we have the following.

**Corollary 1.** *For the Hamming and Cayley metrics,  $\text{GapSDP}_\gamma$  for  $\gamma \geq n/\log n$  is not NP-hard unless coNP has constant round interactive protocols with constant error probability with the verifier allowed  $n^{O(\log n)}$  running time.*

Recall that  $\text{GapMWP}_\gamma$  is Turing reducible to  $\text{GapSDP}_\gamma$  for solvable groups by Theorem 3 and the Turing reduction makes queries with the same gap. Hence, by the above corollary it follows that  $\text{GapMWP}_\gamma$  w.r.t. solvable groups for  $\gamma > n/\log n$  is also unlikely to be NP-hard for Hamming and Cayley metrics.

## References

- [ABSS97] S. ARORA, L. BABAI, J. STERN, E.Z. SWEEDYK, The hardness of approximate optima in lattices, codes, and system of linear equations. *Journal of Computer and System Sciences*, 54(2):317-331. Preliminary version in FOCS’93.
- [AKS01] M. AJTAI, R. KUMAR, D. SIVAKUMAR, A sieve algorithm for the shortest lattice vector. *In Proceedings of the 30th Annual ACM Symposium on Theory of Computing*, 266-275, 2001.
- [BL83] L. BABAI, E.M. LUKS, Canonical labeling of graphs. *Proceedings of the Fifteenth Annual ACM Symposium on Theory of Computing*, pages 171–183, 1983.
- [BCW06] C. BUCHHEIM, P.J. CAMERON, T. WU, On the Subgroup Distance Problem *ECCC*, TR06-146, 2006.
- [CW06] P.J. CAMERON, T. WU, The complexity of the Weight Problem for permutation groups. *Electronic Notes in Discrete Mathematics*, 2006.
- [DH98] M. DEZA, T. HUANG, Metrics on Permutations, a Survey. *J. Combin. Inform. System Sci.* 23,173-185,1998.
- [DMS99] I. DUMER, D. MICCIANCIO, M. SUDAN, Hardness of approximating minimum distance of a linear code. *40th Annual Symposium on Foundations of Computer Science*, 475-484, 1999.
- [GG00] O. GOLDREICH, S. GOLDWASSER, On the limits of nonapproximability of lattice problems. *Journal of Computer and System Sciences*, 60(3):540-563, 2000.
- [GMSS99] O. GOLDREICH, D. MICCIANCIO, S. SAFRA, J. P. SEIFERT, Approximating shortest lattice vector is not harder than approximating closest lattice vectors. *Information Processing Letters*, 71(2):55-61, 1999.
- [Lu93] E.M. LUKS, Permutation groups and polynomial time computations. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, 11:139–175, 1993.
- [MG02] D. MICCIANCIO, S. GOLDWASSER, *Complexity of Lattice Problems. A Cryptographic Perspective*, Kluwer Academic Publishers, 2002.
- [Re] O. REGEV, Lecture Notes - Lattices in Computer Science, [http://www.cs.tau.ac.il/~odedr/teaching/lattices\\_fall\\_2004/index.html](http://www.cs.tau.ac.il/~odedr/teaching/lattices_fall_2004/index.html).