Algorithmic Problems for Metrics on Permutation Groups

V. Arvind, Pushkar S. Joglekar

The Institute of Mathematical Sciences

CIT Campus, Taramani, Chennai 600 113, India

{arvind, pushkar}@imsc.res.in

Current Trends in Theory and Practice of Computer Science

SOFSEM 2008

January 19-25, High Tatras, Slovakia 2008

Introduction.

We study the complexity of Minimum Weight Problem and Subgroup Distance Problem for various metrics over permutation groups.

Road-map of the talk.

- Metrics on permutation Groups
- Minimum Weight Problem and Subgroup Distance Problem
- Elementary hardness result for MWP and SDP and a relation with binary linear codes.
- MWP reduces to SDP for solvable groups

Road-map Contd...

- A $2^{O(n)}$ algorithm for MWP with respect to l_∞ metric
- Other results:
  - Schrier-Sims Algorithm and Minimum Weight Problem with respect to hamming metric
  - Finding fixed point free permutations in $2^{O(n)}$ time
  - Limits of hardness of approximation and a Co-AM protocol for SDP

Metrics on permutation group

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: $d(\pi, \tau) \leq d(\pi, \psi) + d(\psi, \tau)$.

Denote $d(e, \tau)$ by $\|\tau\|$. e is an identity permutation in G.

Examples:

- Hamming distance $d(\tau, \pi) = |\{i|\tau(i) \neq \pi(i)\}|$
- l_∞ distance $d(\tau, \pi) = max_{1 \leq i \leq n}|\tau(i) - \pi(i)|$.
- Cayley distance $d(\tau, \pi)$= minimum number of transpositions taking τ to π

Minimum Weight Problem

Minimum Weight Problem:

Input: (G,k),$G \leq S_n$ given by a generating set and $k > 0$

Question: Is there a $\tau \in G \setminus \{e\}$ with $\|\tau\| \leq k$?

Exact analogue of Shortest Vector Problem in integer lattices [MG02].

Complexity: In general NP-hard for various metrics [CW06].

Subgroup Distance Problem

Subgroup Distance Problem:

Input: (G, τ, k), where $G \leq S_n$ is given by a generating set, $\tau \in S_n$, and $k > 0$

Question: Is $d(\tau, G) \leq k$?

Exact analogue of Closest Vector Problem in integer lattices

Complexity: In general NP-hard for various metrics [BCW06].

MWP and SDP are NP-hard

A simple reduction

- Given $C \subseteq \mathbb{F}_2^n$, a binary linear code. There is an easy way to get an abelian 2-group $G \leq S_{2n}$ isomorphic to additive group of C.
- This implies that MWP and SDP are NP-hard for hamming, l_p metric, Cayley metric using hardness results for analogous problems in codes[ABSS97, DMS99].

MWP Turing reduces to SDP for solvable groups

A Turing reduction from MWP to SDP for solvable groups

- Our reduction uses ideas from [GMSS99] which gives analogous reduction in lattice setting.
- Let $G \leq S_n$ is solvable group. Goal is to check whether a “shortest” permutation in G has norm less than m.
- Obvious approach doesn’t work! The idea is to make different queries of the form (H, τ) to SDP routine for suitable choice of $H \leq G$ and $\tau \notin H$.

MWP Turing reduces to SDP for solvable groups

- Consider the composition series of $G = G_k \triangleright G_{k-1} \triangleright \ldots \triangleright G_1 \triangleright G_0 = \{e\}$, $k \leq n$ such that G_i/G_{i-1} has prime order p_i. Let $\tau_i \in G_i \setminus G_{i-1}$.
- It is easy to see that τ_i's form a generating set of G.
- Query the oracle of SDP for instances $(G_{i-1}, \tau_i^{-r}, m)$ for $1 \leq i \leq k, 1 \leq r < p_i$. Output "YES" if any of the queries outputs "YES" otherwise output "NO".

$2^{O(n)}$ algorithm for MWP (l_∞ metric)

- Given $G \leq S_n$, Goal is to find $\tau \in G \setminus \{e\}$ with minimum norm wrt l_∞ metric
- Brute force algorithm may take $O(n!)$ time
- Our algorithm uses the framework developed in [AKS01], particularly a presentation of AKS algorithm in O. Regev's lecture notes
- The algorithm is randomized and uses $2^{O(n)}$ time and succeeds with probability exponentially close to one.

$2^{O(n)}$ algorithm for MWP (l_∞ metric) Contd..

- $B_n(\tau, r, d) = \{\pi \in S_n | l_\infty(\pi, \tau) \leq r\}$ be the ball of radius r centred at τ. Volume of a ball is number of permutations inside it.
- A volume bound

Lemma 1 For $1 \leq r \leq n-1$ we have,
$r^n/e^{2n} \leq Vol(B_n(e, r, l_\infty)) \leq (2r+1)^n$.
Proof of the Lemma is based on simple combinatorics.

$2^{O(n)}$ algorithm for MWP (l_∞ metric) Contd..

Randomly sampling permutations from l_∞ metric balls

- Pick a function $\tau : [n] \mapsto [n]$ as follows
  For each $i \in [n]$, let $L_i = \{j | 1 \leq j \leq n, i - r \leq j \leq i + r\}$
  and pick $\tau(i)$ uniformly at random from L_i.
- Of course τ defined this way need not be a permutation
- Lemma 1 guarantees that it is so with probability atleast 2^{-cn} !

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

$2^{O(n)}$ algorithm for MWP (l_∞ metric) Contd..

The sieving procedure (Similar to AKS)

- Following is a crucial Lemma used in the algorithm Lemma 3 [Sieving Procedure] Let $r > 0$ and $\{\tau_1, \tau_2, \tau_3, \ldots, \tau_N\} \subseteq B_n(e, r)$ 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 $l_\infty(\tau_i, \tau_j) \leq r/2$.
- The procedure uses simple greedy strategy. Proof of correctness is based on the volume bound in Lemma 1 and a packing argument.

$2^{O(n)}$ algorithm for MWP (l_∞ metric) Contd..

Main Algorithm

- we can assume that we know a norm of a “shortest” permutation τ. Let $t = \|\tau\|$.
- Let $N = 2^{cn}$. For $1 \leq i \leq N$, pick ρ_i independently and uniformly at random from G, and pick τ_i almost uniformly at random from $B_n(e, 2t)$.
- Let $\psi_i = \tau_i \rho_i$, $1 \leq i \leq N$. Let $Z = \{(\psi_1, \tau_1), (\psi_2, \tau_2), \ldots, (\psi_N, \tau_N)\}$, and let $R = max_i \|\psi_i\|$. Let $T = [N]$.

$2^{O(n)}$ algorithm for MWP (l_∞ metric) Contd..

While $R > 6 * t$ do the following steps:

- Apply the “sieving procedure” to $\{\psi_i \mid i \in T\}$. Let $S \subseteq T$ be the output of sieving procedure.
- for all $i \in S$ remove tuple (ψ_i, τ_i) from Z.
- 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 $l_\infty(\psi_j, \psi_i) \leq R/2$.
- set $R = R/2 + 2t$ and $T = T \setminus S$.
- 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.

$2^{O(n)}$ algorithm for MWP (l_∞ metric) Contd..

Proof of correctness

- Invariant maintained through out the algorithm : For all $i \in T$ we have $(\varphi_i, \tau_i) \in Z$, $\tau_i^{-1}\varphi_i \in G$ and $\|\varphi_i\| \leq R$.
- From Lemma 3 if follows that only “few” elements are sieved out in the while loop
- As a result we have $2^{O(n)}$ tuples (φ_i, τ_i) such that $\tau_i^{-1}\varphi_i \in G$ and $\|\tau_i^{-1}\varphi_i\| \leq 8t$

$2^{O(n)}$ algorithm for MWP (l_∞ metric) Contd..

Proof of correctness

- We have $2^{O(n)}$ permutations in G with small norms, so we already have a good approximation!
  Are we through?
- Not really ! all of them can be identity permutations !
- We can argue that we get not even the approximation to "shortest permutation" but can find it exactly
  At this point our proof crucially differs from that of [Re]

Other Results

- A $2^{O(n)}$ algorithm for MWP (hamming metric) using Schrier-Sims algorithm
- A $2^{O(n)}$ algorithm for Finding fixed point free permutation
- A Co-AM protocol for SDP

Future Work

- Using ideas similar to $2^{O(n)}$ algorithm for MWP (l_∞ metric) we could get a $2^{O(n)}$ algorithm for solving gap version of SDP for gap $1+\epsilon$.
- In case of integer lattices for certain problems a worst-case to average case reduction in known . Interesting direction of further research would be to explore the possibility of worst-case to average case reduction in permutation group setting. Interestingly AKS algorithm uses ideas from Ajtai's work on worst-case to average case reduction.

THANK YOU!!