Approximation of Geodesics in Metabelian Groups

Olga Kharlampovich, Atefeh Mohajeri
(McGill University)

Barcelona, 2011

Abstract

We study the *geodesic problem* in the restricted wreath product of a finitely generated group with a finitely generated abelian group containing $\mathbb{Z}^2$. We prove that the *geodesic problem* is *NP*-hard in this case. In the second part we show that there exists a *Polynomial-Time Approximation Scheme* for this problem. Existence of a 2-approximation algorithm for the *geodesic problem* in free metabelian groups will be shown in the third part.

The Geodesic Problem (GP) : Let G be a finitely generated group and X be a set of generators of G. The *Geodesic Problem* relative to X is the following problem: given a word $g \in G$ find a word $w \in F(X)$ with a minimal length such that $g = w$ in G. We define $l_X(g)$ the geodesic length of g to be $|w|$.
The Geodesic Length Problem (GLP) Let G be a finitely generated group and X be a set of generators of G. Given a word $w \in F(X)$ find $l_X(w)$
Bounded Geodesic Length Problem (BGLP) Let G be a finitely generated group with a set of generators X. Given a word $w \in F(X)$ and $k \in \mathbb{N}$ determine if $l_X(w) \leq k$.

Miasnikov, Romankov, Ushakov, Vershik show that *BGLP* is *NP*-complete for free metabelian groups of arbitrary rank. Since *BGLP* is polytime reducible to the Geodesic Length problem(*GLP*), *GLP* is *NP*-hard in free metabelian groups.

They posed the problem to describe wreath products of two f.g. abelian groups that have GP in P.

Polynomial-time Reducibility Problem P is Polynomial-time reducible to problem Q if whenever P has a solution, such a solution can be obtained from a solution of Q in polynomial time.
Class NP Class NP is the class of all decision problems for which the "yes"-instances are recognizable in polynomial time by a non-deterministic turing machine= *easily verifiable*
NP-complete A problem P is NP-complete if it is in class NP and any other problem in class NP is reducible to P in polynomial-time. In other words, P is at least as hard as any other problem in class NP.
NP-hard A problem P is NP-hard if any NP-complete problem is polynomial time Turing reducible to P.

Let A and B be groups. Let

$$D = \{f : B \mapsto A || sup(f)| \leq \infty\}$$

$$\hat{b} : D \mapsto D$$
$$f(.) \mapsto f(b^{-1}.)$$

Thus, we have an embedding $B \hookrightarrow Aut(D)$ by identifying b with $\hat{b}$. The restricted wreath product $G = A \wr B$ is the restricted semidirect product $D \rtimes B$. So, G consists of all pairs (f, b), $f \in D$ and $b \in B$, equipped with the operation

$$(f, b)(f', b') = (fb(f'), bb').$$

We identify $a \in A$ with f_a (the function mapping identity to a and everything else to the identity).

Let B be abelian. Let $a_{i_j} = a_1^{m_{j1}} \dots a_t^{m_{jt}}$ for $0 \leq j \leq k$, $h_i = b_1^{n_{i1}} \dots b_r^{n_{ir}}$ for $1 \leq i \leq k$ and $h' = b_1^{n'_1} \dots b_r^{n'_r}$. Then, $g = h_1 a_1 \dots h_k a_k h'$. We can rewrite g as follows (using the fact that B is abelian)

$$\begin{aligned} g =& (h_1 a_1 h_1^{-1})(h_1 h_2 a_2 h_2^{-1} h_1^{-1})(h_1 h_2 h_3 a_3 h_3^{-1} h_2^{-1} h_1^{-1}) \\ & \dots (h_1 h_2 \dots h_k a_k h_k^{-1} \dots h_2^{-1} h_1^{-1}) h_1 \dots h_k h' \\ =& a_1^{h_1^{-1}} a_2^{h_1^{-1} h_2^{-1}} \dots a_k^{h_1^{-1} \dots h_k^{-1}} h' h_1 \dots h_k \end{aligned}$$

For a given presentation of g:

$$g = b_1^{n_{11}} \dots b_r^{n_{1r}} a_1 b_1^{n_{21}} \dots b_r^{n_{2r}} a_2 \dots b_1^{n_{k1}} \dots b_r^{n_{kr}} a_k b_1^{n'_1} \dots b_r^{n'_r}$$

we denote by l the number of generators used in this particular presentation of g assuming that we use $|a_i|_A$ generators for a_i. Then,

$$l = \sum_{i=1}^{r} n_{1i} + \ldots + \sum_{i=1}^{r} n_{ki} + \sum_{i=1}^{r} n'_i + \sum_{i=1}^{k} |a_i|_A$$

We want to relate this sum to a path in the Cayley graph of B. Let $C = \mathbb{Z}^r$ viewed as the Cayley graph of B with respect to the set of generators S_B. Let v_0 be the vertex of C corresponding to identity. Let $x = b_1^{x_1} \dots b_r^{x_r}$ and $y = b_1^{y_1} \dots b_r^{y_r}$ be two vertices in C. The *Manhattan distance* between x and y is defined as

$$d(x, y) = \sum_{i=1}^{r} |x_i - y_i|.$$

By a walk $w_1, w_2, \dots w_r$ in C we mean a path going from w_i to w_{i+1}, at each step taking a shortest path between these two points (a path whose length is $d(w_i, w_j)$).

Let $g = a_1^{w_1} \dots a_k^{w_k} b$ where w_i and b were introduced previously. Consider the walk $v_0, w_1, w_2, \dots w_k, b$ in C. While going from v_0 to w_1, the distance traveled is $\sum_{i=1}^{r} n_{1i}$ with respect to Manhattan distance. In the next step, we go from

$$w_1 = h_1^{-1} = b_1^{-n_{11}} \dots b_r^{-n_{1r}}$$

to

$$w_2 = h_1^{-1} h_2^{-1} = b_1^{-n_{11}-n_{21}} \dots b_r^{-n_{1r}-n_{2r}}$$

.

Thus, the distance traveled this time is $\sum_{i=1}^{r} n_{2i}$. By the same observation, the length of the path taken by the walk between w_j and w_{j+1} is $\sum_{i=1}^{r} n_{ji}$. In the last step, going from w_k to b gives us $\sum_{i=1}^{r} n'_i$ term.

As it was mentioned before we can assume that $w_i \neq w_j$ for $i \neq j$. Thus, $a_i^{w_i}$ commutes with each other.

If we flip $a_1^{w_1}$ and $a_2^{w_2}$, we get

$$\begin{aligned} g &= a_2^{w_2} a_1^{w_1} \dots a_k^{w_k} b \\ &= w_2^{-1} a_2 w_2 w_1^{-1} a_1 w_1 w_3^{-1} a_3 w_3 w_4^{-1} \dots w_k^{-1} a_k w_k b \\ &= b_1^{n_{11}+n_{21}} \dots b_r^{n_{1r}+n_{2r}} a_2 b_1^{-n_{21}} \dots b_r^{-n_{2r}} a_1 b_1^{n_{21}+n_{31}} \\ &\quad \dots b_r^{n_{2r}+n_{3r}} a_3 b_1^{n_{41}} \dots b_r^{n_{4r}} \dots b_1^{n_{k1}} \dots b_r^{n_{kr}} a_k b_1^{n_1'} \dots b_r^{n_r'} \end{aligned}$$

The number of generators we get this time is

$$\sum_{i=1}^{r}(n_{1i}+n_{2i})+\sum_{i=1}^{r} n_{2i}+\sum_{i=1}^{r}(n_{2i}+n_{3i})+\sum_{i=1}^{r} n_{4i} \dots+\sum_{i=1}^{r} n_{ki}+\sum_{i=1}^{r} n_i'+\sum_{i=1}^{k} |a_i|$$

The first term corresponds to going from v_0 to w_2 in C. The second and the third terms represent the walks w_2, w_1 and w_1, w_3 respectively. The other terms represent the walk $w_3, w_4, \dots w_k, b$. Flipping $a_i^{w_i}$, corresponds to different walks in C which start at v_0 ,go through all w_i in different order and stop at b in the last step.

Definition

Let $S = \{w_1, \dots, w_m, b\}$ be a set of elements of B. We say a walk l in C is a Hamiltonian walk on S with end point b if l is a walk of the form $v_0, w_{i_1}, \dots, w_{i_m}, b$, where $w_{i_1}, \dots, w_{i_m}$ is a permutation of $w_1, \dots, w_m$. We call such l minimum if it has the minimum length among all such paths.

The observation made above suggest the following result which was first proved by W. Parry.

Lemma

Let $g = a_1^{w_1} \ldots a_k^{w_k} b \in G$ where w_i are as above and all are different. Any geodesic word g' representing g corresponds to a minimum Hamiltonian walk in C on $S = \{w_1, \ldots w_m, b\}$ with end point b.

Corollary

If we know how to solve GP in A, solving GP in $G = A \wr B$ is equivalent to finding a minimum Hamiltonian walk in a subgraph of the cayley graph of B.

The Traveling Salesman Problem in $\mathbb{Z}^r$ Given a set of n points in $\mathbb{Z}^r$ and a metric on $\mathbb{Z}^r$, find a minimum Hamiltonian cycle in the complete graph on this set of points, i.e. a minimum cycle which visits all points exactly once.

It has been shown that the *Traveling Salesman Problem* in $\mathbb{Z}^r$ with *manhattan distance* is NP-hard for $n \geq 2$ (Papadimitriou, Garey et al.).

Theorem Geodesic Problem in $G = A \wr B$ is NP-hard where A is a finitely generated group and B is a finitely generated abelian group containing $\mathbb{Z}^2$.

Remark The Bounded Geodesic Length Problem($BGLP$) is the decision version of the Geodesic problem. An immediate consequence of the previous theorem is that if $BGLP$ for A is in class NP then it is NP-complete in G.

Theorem If Geodesic Problem in A is polynomial and $B = \mathbb{Z} \times \mathbb{Z}_{k_1} \times \ldots \mathbb{Z}_{k_t}$, then Geodesic Problem in $G = A \wr B$ is polynomial.

Proof A polynomial time algorithm to solve TSP in the Cayley graph of $B = \mathbb{Z} \times \mathbb{Z}_{k_1} \times \ldots \mathbb{Z}_{k_t}$ with the degree of the polynomial equal to $k_1 \ldots k_t$ can be constructed. There exists even better algorithm for TSP in B with the degree of the polynomial independent on $k_1 \ldots k_t$ (Babai).

Approximation Algorithm. Let denote by OPT the optimal solution of an minimization problem P . An algorithm A approximates problem P within a factor of $\rho \geq 1$ if $f(A)/OPT \leq \rho$, where $f(A)$ is the solution given by A .

Polynomial-Time Approximation Scheme . A $PTAS$ for a problem is a family of polynomial-time algorithms such that for any given constant c, there is an algorithm in the family that approximates the problem within a factor of $(1 + 1/c)$. The running time of the algorithm might depend on c, but for each fixed c, it is polynomial in the size of the input.

Theorem

There exists a PTAS for the Geodesic Problem in G, assuming the Geodesic Problem in A is solvable in polynomial time.

Proposition

(Arora, 96) There exists a PTAS for TSP in R^d with Euclidean norm which generalizes to other L^p norms for $p \geq 1$. A randomized version of the algorithm gives an approximation within a factor of $(1 + 1/c)$ of the optimal tour in $\mathcal{O}(n(logn)^{(\mathcal{O}(\sqrt{(}dc)))^{d-1}})$. If we derandomize the algorithm, we multiply the running time by $\mathcal{O}(n^d)$.

Proposition

The same is true if instead of a TSP tour we consider TSP path.

Let $F = F(X)$ be a free group of rank r. Denote by $F' = [F, F]$ the derived subgroup of F and by $F'' = [F', F']$ the second derived subgroup of F. Let $G = F/F'$ and $H = F/F''$. G is a free abelian group of rank r and H is a free metabelian group of rank r.

Let p be a path in a directed graph γ from v_1 to v_2 and let $\pi_p : E \rightarrow \mathbb{Z}$ be such that $\pi_p(e)$ is equal to the number of times that p passes through e counted -1 when p takes e backward. π_p satisfies the flow condition for v_1 as source and v_2 as sink .

Let $\mu' : F(X) \to H$ be the canonical epimorphism.

In Minimum Steiner Tree Problem we are given a graph $\gamma = (V, E)$ and a subset $V_1 \subset V$. We need to find a minimum subgraph(subtree) such that covers all vertices in V_1. In Minimum group Stiener Tree Problem we are given connected components C_i of γ and we want to find a minimum subgraph that makes the subgraph $\cup C_i$ connected.
Let $w \in G$ and π_w be the flow induced by w in $\gamma = Caley(G)$. We define $supp(\pi_w) = \{e \in E | \pi(e) \neq 0\}$. A minimum Group Steiner Tree for w is a minimal Group Steiner Tree for the connected components of the subgraph induced by $supp(\pi_w)$ in γ.

Miasnikov, Romankov, Ushakov, Vershik show that $BGLP$ is NP-complete for free metabelian groups of arbitrary rank. Since $BGLP$ is polytime reducible to the Geodesic Length problem(GLP), GLP is NP-hard in free metabelian groups. We show that there is a 2-approximation for the geodesic problem in H.

Proposition

(M., R., U., V.) Let γ be the cayley graph of G with respect to the set of generators $\mu(X)$. then for $w \in F$ we have

$$l_X(\mu'(w)) = \sum_{e \in supp(\pi_w)} \pi_w(e) + 2|E(Q)| \qquad (1)$$

where Q is a minimum group steiner tree for w in γ and π_w is the flow induced by w in γ.

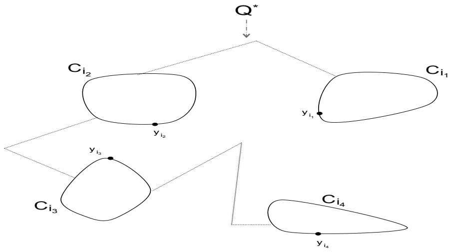
In the expression for $l_X(\mu'(w))$ we can evaluate $\sum_{e \in \text{supp}(p_w)} \pi_w(e)$ in polynomial time. The time consuming part is finding Q. Unfortunately, there has been no *PTAS* known so far for the *Group Steiner Tree* problem. Nevertheless, in the same Arora'a paper a similar statement to the *PTAS* result for *Euclidean Traveling Salesman Problem* is proved for *Minimum Steiner Tree Problem*.

Proposition

(Arora) There exists a PTAS for the Euclidean Minimum Steiner Tree in R^d *which generalizes to other* L^p *norms for* $p \geq 1$*. A randomized version of the algorithm gives an approximation within a factor of* $(1+1/c)$ *of the optimal tree in* $\mathcal{O}(n(logn)^{(\mathcal{O}(\sqrt{(}dc)))^{d-1}}$*. If we derandomize the algorithm, we multiply the running time by* $\mathcal{O}(n^d)$*.*

Theorem

There is a 2-approximation polynomial algorithm for the Geodesic Problem in free metabelian groups of arbitrary rank r.



PTAS for TSP in $\mathbb{R}^2$ with ℓ_2 norm. The algorithm uses divide-and-conquer, and the "divide" part is a randomized version of the classical quadtree, which partitions the instance using squares that progressively get smaller

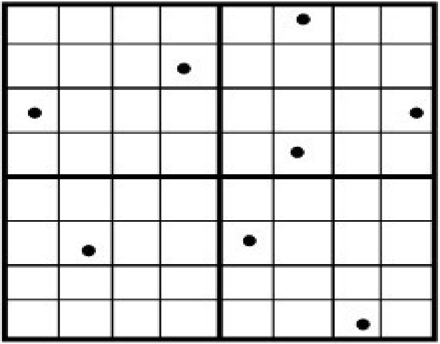


Fig. 1. The dissection

Although the algorithm is described as randomized, it can be derandomized with some loss in efficiency, specifically, by trying all choices for the shifts used in the randomized dissection.

First we perform a simple perturbation of the instance that, without greatly affecting the cost of the optimum path, ensures that each node lies on the unit grid (i.e., has integer coordinates) and every internode distance is at least 2. Call the smallest axis-parallel square containing the nodes the bounding box. Our perturbation will ensure its side length is at most $n^2/2$.

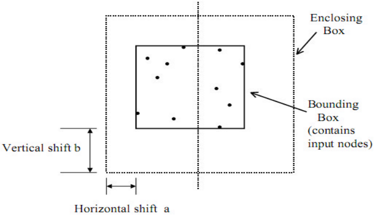


Fig. 2. The enclosing box contains the bounding box and is twice as large, but is shifted by random amounts in the x and y directions.

Here a, b are random.

If L is the side length of the enclosing box, we assign a level from 0 to $\log L - 1, \ldots, 1$ to each horizontal and vertical grid line that participated in the dissection. The horizontal (resp., vertical) line that divides the enclosing box into two has level 0. Similarly, the 2^i horizontal and 2^i vertical lines that divide the level i squares into level $i+1$ squares each have level i.

Portal-respecting paths Each grid line will have special points on it called portals. A level i line has $2^{i+1}m$ equally spaced portals inside the enclosing box, where m is the portal parameter (to be specified later). In addition, we also refer to the corners of each square as a portal. Since the level i line has 2^{i+1} level $i+1$ squares touching it, we conclude that each side of the square has at most $m+2$ portals. A portal-respecting path is one that, whenever it crosses a grid line, does so at a portal.

A portal-respecting path is k-light if it crosses each side of each dissection square at most k times. The optimum portal-respecting tour does not need to visit any portal more than twice; this follows by the observation that removing repeated visits can, thanks to triangle inequality, never increase the cost. Thus the optimum portal-respecting tour is $(m+2)$-light. A simple dynamic programming can find the optimum portal-respecting tour in time $2^{O(m)} L log L$.

The dynamic programming. Suppose we are interested in portal-respecting paths that enter, exit each dissection square at most $4k$ times. The subproblem inside the square can be solved independently of the subproblem outside the square so long as we know the portals used by the tour to enter, exit the square, and the order in which the tour uses these portals. Note that given this interface information, the subproblems inside and outside the square involve finding not salesman paths but a set of up to $4k$ vertex-disjoint paths that visit all the nodes and visit portals in a way consistent with the interface.

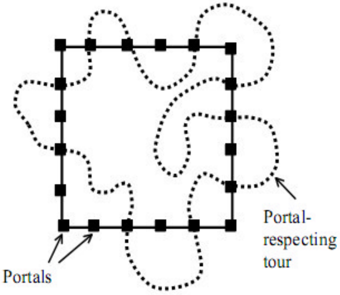


Fig. 3. This portal-respecting tour enters and leaves the square 10 times, and the portion inside the square is a union of 5 disjoint paths.

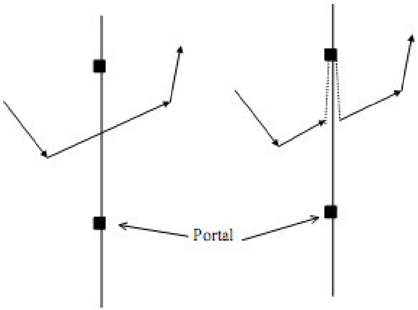


Fig. 4. Every crossing is moved to the nearest portal by adding a "detour."

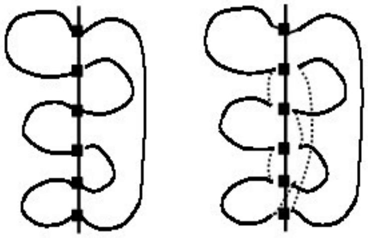


Fig. 5. The tour crossed this line segment 6 times, but breaking it and reconnecting on each side (also called "patching") reduced the number of crossings to 2.