

# Euclidean Movement Minimization

Nima Anari\*    MohammadAmin Fazli†    Mohammad Ghodsi†‡    Pooya Jalaly Khalilabadi†  
 MohammadAli Safari†§

## Abstract

We consider a class of optimization problems called *movement minimization on euclidean plane*. Given a set of nodes on the plane, the aim is to achieve some specific property by minimum movement of the nodes. We consider two specific properties, namely the connectivity (CON) and realization of a given topology (TOPOL). By minimum movement, we mean either the sum of all movements (SUM) or the maximum movement (MAX). We obtain several approximation algorithms and some hardness results for these four problems. We obtain an  $O(m)$ -factor approximation for CONMAX and CONSUM and an  $O(\sqrt{m/OPT})$ -factor approximation for CONMAX. We also extend some known result on graphical grounds in [1, 2] and obtain inapproximability results on the geometrical grounds. For the TOPOL problem (where the final decoration of the nodes must correspond to a given configuration), we find it much simpler and provide FPTAS for both MAX and SUM versions.

## 1 Introduction

Consider a number of moveable robots distributed over a plane in a far-flung manner. Each robot has an antenna with a limited maximum range, denoted by  $r_{max}$ . Robot  $s$  can communicate directly with robot  $t$  if and only if their distance is less than  $r_{max}$ . Robot  $s$  can also communicate indirectly with  $t$  if there is an ordered set of robots  $s = r_1, r_2, \dots, r_p = t$  so that each  $r_i$  can directly communicate with  $r_{i+1}$ . With this explanation, we can form a dynamic graph whose vertices are the moveable robots on the plane and edges are formed by connecting each robot to every other robot residing in the disk with radius  $r_{max}$  around it. These geometric graphs are called UDGs (Unit Disk Graphs).

\*Computer Science Division, University of California Berkeley, email: anari@cs.berkeley.edu

†Department of Computer Engineering, Sharif University of Technology, emails: fazli, jalaly@ce.sharif.edu, ghodsi, safari@sharif.edu

‡Institute for Research in Fundamental Sciences (IPM), Tehran, Iran. This author’s research was partially supported by IPM under grant No: CS1389-2-01

§The research was partially supported by the Institute for Research in Fundamental Sciences under grant No: CS1389-4-09

**Definition 1** *Given some points  $p_1, \dots, p_m$  in the euclidean plane, the UDG on these points is defined as a simple graph  $G = (V, E)$ , where  $V = \{1, \dots, m\}$  and  $E = \{\{i, j\} \mid |p_i - p_j|_2 \leq 1\}$*

Suppose that robots are initially located at points  $p_1, p_2, \dots, p_m$ . It is clear that all robots can communicate directly or indirectly with each other if and only if their corresponding UDG is connected. Our aim is to have the robots move in a way that they form a connected UDG after relocation (the points  $p_1^*, p_2^*, \dots, p_m^*$ ). We also want to efficiently optimize the travel distance of the robots before they reach their final locations. The term *efficiently* can be defined in many ways. In this paper, we consider two of such measures: namely SUM and MAX. In SUM, the goal is to minimize the sum of the movements of all robots, or formally to minimize  $\sum_{i=1}^m |p_i^* - p_i|_2$ . This parameter roughly measures the total energy consumed by the robots. In MAX, the goal is to minimize the maximum movement of all robots, i.e. minimizing  $\max_{i \in \{1, \dots, m\}} |p_i^* - p_i|_2$ . This parameter measures the amount of time needed to reach the final locations.

Using these two functions, we define two problems: CONMAX and CONSUM.

**Definition 2** *In CONMAX (resp. CONSUM) we want to move the robots so as to form a connected UDG and the optimization goal is MAX (resp. SUM).*

Each of these problems can be considered in both graphical or geometrical settings.

**Definition 3** *In a graphical setting, robots move on a graph. At first, robots are placed on some vertices of the graph and at each turn, each robot can move to one of the adjacent vertices (each edge is considered to have one unit of length). In geometrical settings, robots are points belonging to a geometrical space ( $\mathbb{R}^2$  in this paper) and are free to move in any direction in the space.*

## 1.1 Other Works

Demaine et al. [1, 2] first introduced movement problems in graphical settings and extensively studied them. They defined 15 types of movement problems (borrowing from their terminology, from here on we use the

words robot and *pebble* interchangeably). They consider five properties: connectivity, directed connectivity, path, independent set and matching and consider three objective functions: maximum movements, total movement and number of pebbles that move. This results in the following 15 problems: CONMAX, CONSUM, CONNUM, DIRCONMAX, DIRCONSUM, DIRCONNUM, PATHMAX, PATHSUM, PATHNUM, INDMAX, INDSUM, INDNUM, INDMAX, INDSUM, INDNUM.

Most of their salient results were proven in the context of graphs. They proposed an  $\mathcal{O}\left(\sqrt{\frac{m}{OPT}}\right)$ -factor approximation algorithm for CONMAX and PATHMAX ( $m$  is the number of pebbles) and proved  $\Omega(n^{1-\epsilon})$  inapproximability result for CONSUM and DIRCONMAX ( $n$  is the number of vertices in the ground graph) in graphical settings. They also gave an  $\mathcal{O}(1)$ -approximation for INDMAX with an additive error of  $\mathcal{O}(1)$  in geometrical settings.

Note that all the algorithms presented in [1, 2] are in fact polynomial in  $n$ , the number of the nodes in the base graph, which makes them inefficient when  $n \gg m$  which is a realistic assumption. Dealing with this, given that the number of mobile agents is typically much smaller than the complexity of the environment, in [3] the authors turn to fixed-parameter tractability. They characterize the boundary between tractable and intractable movement problems in a very general set up and show that many movement problems of interest have fixed parameter tractable algorithms.

## 1.2 Our Results

Our results include algorithms for CONMAX, CONSUM, TOPOLMAX, TOPOLSUM and an inapproximability result for CONMAX.

In section 2.1 we prove  $(2 - \frac{\sqrt{2}}{2})$ -inapproximability for CONMAX in geometric settings which extends the hardness result of Demaine et al. [1, 2] about CONMAX in graphical settings.

**Theorem 1** *There is no polynomial algorithm for CONMAX in geometrical settings with an approximation factor of less than  $2 - \frac{\sqrt{2}}{2}$ , unless  $\mathcal{P} = \mathcal{NP}$*

In section 2.2 and 2.3 we give approximation algorithms for CONMAX and CONSUM on geometrical grounds. We present  $\mathcal{O}(m)$ -factor approximation algorithm for both problems which improve the  $\mathcal{O}\left(\sqrt{\frac{m}{OPT}}\right)$ -approximation algorithm (with additive error of  $\mathcal{O}(1)$ ) of Demaine et al. [2] in the cases where  $OPT$  is very small.

**Theorem 2** *There is an  $\mathcal{O}(m)$ -factor approximation algorithm for CONMAX and CONSUM on geometrical grounds.*

In the final part of this paper, we introduce a new kind of movement problems which is more constrained, in some sense, than the previously proposed problems: TOPOLMAX and TOPOLSUM.

**Invariant 1** *In problems TOPOLMAX and TOPOLSUM, we are given  $m$  initial points  $p_1, \dots, p_m \in \mathbb{R}^2$  and a set of edges  $E \subseteq \{\{i, j\} \mid i, j \in \{1, \dots, m\}\}$ . We are supposed to determine  $m$  points  $p_1^*, \dots, p_m^* \in \mathbb{R}^2$  in such a way that the UDG defined on  $p_1^*, \dots, p_m^*$  contain all of the edges in  $E$ . The objective function we are trying to minimize can be either MAX or SUM which results in two different problems we call TOPOLMAX and TOPOLSUM.*

Although our results are stated in two dimensions, most of them can be easily extended to higher dimensions. In particular all of our approximation algorithms work for higher dimensions too.

**Theorem 3** *There is a FPTAS for the problems TOPOLMAX and TOPOLSUM.*

## 2 CONMAX and CONSUM

### 2.1 Hardness Results

In this section we prove Theorem 1. First, we prove that CONMAX is 2-approximable on UDGs (graphical ground) only if  $\mathcal{P} = \mathcal{NP}$ . Then, with minor modifications, we prove Theorem 1. Our main idea is a proof of Demaine et al. in [1, 2] for hardness of CONMAX problem in graphical settings, but our case is more involved and needs many modifications.

For this, we reduce the hamiltonian cycle problem on 3-regular planar graphs which is known to be  $\mathcal{NP}$ -hard [7]. Let us call this problem 3PHP.

We first start with a useful way of embedding planar graphs:

**Lemma 4** (Valiant [6]). *A planar graph  $G$  with maximum degree 4 can be embedded in the plane using  $\mathcal{O}(|V|)$  area in such a way that its vertices are at integer coordinates and its edges are drawn so that they are made up of line segments of the form  $x = i$  or  $y = j$ , for integers  $i$  and  $j$ .*

There is also a polynomial time algorithm to compute such an embedding [8].

We are now ready to prove the hardness of the CONMAX problem on UDG grounds.

**Theorem 5** *There is no polynomial algorithm for CONMAX on UDG graphical grounds with approximation factor less than 2 unless  $\mathcal{P} = \mathcal{NP}$*

**Proof.** We prove this by reducing 3PHP. Assume that we have an instance of 3PHP problem; a 3-regular planar graph  $G$  in which we want to check for the existence

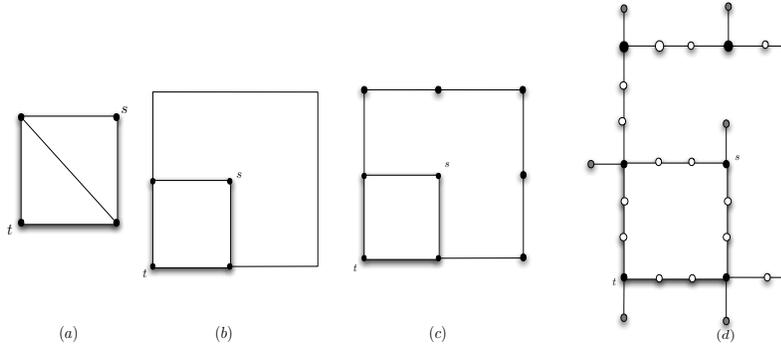


Figure 1: Graph  $G$  and its transformation process.

of a hamiltonian path between two specified vertices  $s$  and  $t$ . See Figure. 1(a) for an example.

First we use Lemma 4 to get an embedding  $H$  of  $G$  with integer coordinate vertices and horizontal or vertical edges (Figure. 1(b)).

Next we scale up all vertex coordinates by 6.0 to make each edge six times longer. The length of every edge  $e = (u, v)$  is now a multiple of 6.0. We put new vertices on every integer-coordinate point between  $u$  and  $v$ . So, the edge  $e = (u, v)$  is replaced by a path  $P^e = u = v_0, v_1, \dots, v_{6k-1}, v_{6k} = v$  (notice that the distance between  $v_i$  and  $v_{i+1}$  is exactly one).

We color vertices  $u = v_3, v_6, \dots, v_{3i}, \dots, v_{6k-3}, v_{6k} = v$  as black and the remaining vertices as white. See the resulting graph  $G'$  in Figure. 1(d) (in this figure we have scaled up everything by 3 and not 6 for clarity and better understanding).

Since the degree of each vertex in the resulting UDG is at most 3, we can attach a new leaf to each black vertex via a unit length vertical or horizontal edge. We color these new leaves as gray and call the resulting new UDG  $G'$ .

Finally, we place one pebble on  $s$  and  $t$  and each gray vertex of  $G'$ . We also place two pebbles on each black vertex of  $G'$  except  $s$  and  $t$ . We show that  $G$  has a hamiltonian path between  $s$  and  $t$  if and only if the answer of CONMAX on  $G'$  is 1. If there is a hamiltonian path between  $s$  and  $t$  in  $G$ , we can move the pebble on each gray vertex to its neighboring vertex in  $V(G')$  and move pebbles on each black vertex to its neighboring vertices along the path corresponding to  $G$ 's hamiltonian path that induce a connected subgraph in  $G'$ . For the reverse side, we show that if  $G$  does not have a hamiltonian path between  $s$  and  $t$  then the value of CONMAX is at least 2.0.

We show that when  $G$  is not hamiltonian, then establishing connectivity in  $G'$  requires a pebble in a gray vertex to move to a white vertex which requires a movement of 2.0.

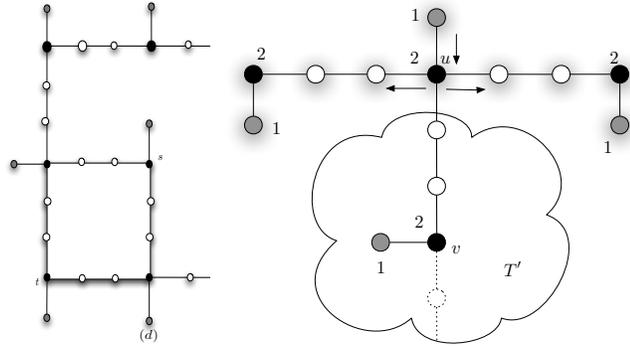


Figure 2: The 3-degree vertex in minimum maximum degree spanning tree viewed in  $G'$ .

Consider the optimal connectivity establishment in  $G'$ . This induces a connected subgraph of  $G$  which is not a hamiltonian path and, therefore, has a maximum degree at least 3. Let  $u$  be a vertex with degree 3. It has only 2 pebbles. So, one of its neighboring white vertices, say  $v$ , can not be covered by the pebbles on it. If we remove the edge between  $u$  and  $v$ , we would have a subtree  $T'$  in which we need at least 2 moves to connect its pebbles to  $u$ 's pebbles. This completes the proof. It is clear that nothing would be changed in this proof if we replace general UDG graphs with their specific type grids because we used only vertical/horizontal edges and integer coordinated vertices.  $\square$

We can also use the above proof for the  $(2 - \frac{\sqrt{2}}{2})$ -inapproximability of CONMAX on geometrical grounds.

**Proof.** (of Theorem 1) The proof structure is almost identical to the proof of Theorem 5. In Figure. 2, the distance between vertex  $u$  and vertex  $v$  is 3. In proof of Theorem 1 we had to move pebbles only in integer units of length and the  $uv$  path was not covered by the pebbles placed on  $u$ . So to connect  $T'$ 's pebbles to  $u$ 's pebbles, we had to move them 2 units and the approximation factor was at least 2.

This is different from geometrical ground as  $u$ 's two pebbles can move to every point of the plane without any limitation. So, there would be a movement of them in which the minimum coverage of these pebbles over all 3 outgoing paths of  $u$  is  $\frac{\sqrt{2}}{2}$  (For example when they move in north-west and south-east direction with 45 degree slope). So the maximum of minimum coverage over these 3 paths by  $u$ 's pebbles is at most  $\frac{\sqrt{2}}{2}$  and again suppose that this minimum coverage is being happened for  $uv$  path. This completes the proof because in this situation the movement of  $T'$ 's pebbles would be at least  $3 - 1 - \frac{\sqrt{2}}{2} = 2 - \frac{\sqrt{2}}{2}$  and this leads to the approximation factor  $2 - \frac{\sqrt{2}}{2}$ .  $\square$

## 2.2 $O(m)$ approximation for CONMAX

If two pebbles are adjacent at the end then their original distance should be at most  $\lambda = 2OPT + 1$ . This means that if we scale down all distances by a factor of  $\frac{1}{\lambda}$  then the corresponding UDG would be connected. This is the idea behind the algorithm: centered at one of the points, scale down all distances by a factor  $\frac{1}{\lambda}$  and move every point to its new location after scaling. This yields an  $O(m)$ -factor approximation. The rest of details is left to the journal version of this paper.

## 2.3 $O(m)$ approximation for CONSUM

We construct a complete weighted graph in which the weight of  $(i, j)$  is defined as  $\max(0, (|p_i - p_j|_2 - 1)/2)$ . Then we find a Minimum Spanning Tree (MST) of this graph. It can be shown that two fifths the weight of the MST is a lower-bound for the optimum solution.

Next, we do the following operation for each edge  $(i, j)$  of the MST: Removing the edge gives us two connected components. We translate each connected component along the edge  $p_i p_j$  by a distance of  $\max(0, (|p_i - p_j|_2 - 1)/2)$ . Note that among the edges of the tree, only the distance between  $p_i$  and  $p_j$  is changed.

After all these operations, all edges of the MST become present in the resulting UDG. The total sum of movements is at most  $m$  times the weight of the original MST, hence an  $O(m)$ -approximation. More details are left to the journal version of this paper.

## 3 Predetermined Topology

Assume that we are given  $m$  different points  $p_1, \dots, p_m \in \mathbb{R}^2$ . The goal is to make the UDG defined on these points have certain properties. One of the properties that might be desirable for the UDG to have, is to have it contain a certain predetermined graph.

Clearly one can assume that the given topology  $E$  is connected; otherwise, the problem can be solved for each connected component separately, and the solutions can be combined together. Hence, from now on we will assume that  $E$  is connected.

The main result we obtain is that there is a FPTAS for each one of these problems. Our FPTAS's use the ELLIPSOID method as a blackbox.

**Remark 1** *The ELLIPSOID method works with a separation oracle defined on a convex set; that is an oracle which when given a point  $p$  determines whether it's inside the convex set, and if the answer is false, returns a hyperplane separating the point and the convex set. Given a convex body  $C \subset \mathbb{R}^n$  and an initial ellipsoid  $E_0$  containing  $C$ , and an arbitrary positive number  $\underline{V}$ , the ELLIPSOID method either finds a point in  $C$ , or finds out that the volume of  $C$  is less than  $\underline{V}$ . The time it*

*takes for the ELLIPSOID method to run is bounded by a polynomial in  $n$  and  $\log(\text{Vol}(C)/\underline{V})$ .*

## 3.1 FPTAS for TOPOLMAX

In this section we will show how TOPOLMAX can be approximated using the ELLIPSOID method.

Our algorithm uses some of the results and tools from the  $O(m)$  approximation algorithm for CONMAX, including the definition and properties of the geometrical transformation homothety. For details refer to the journal version of this paper.

For two given points  $p_i$  and  $p_j$  to become at most 1 unit apart (in the Euclidean metric), one should be moved by at least  $e_{ij} = \max(0, (|p_i - p_j|_2 - 1)/2)$ .

Now given an instance of TOPOLMAX define  $\underline{Q}$  to be  $\max_{\{i,j\} \in E} e_{ij}$ . Clearly  $\underline{Q}$  is a lower-bound for  $OPT$ .

**Lemma 4** *An instance of TOPOLMAX can be solved by a sum of displacements of  $2(m-1)\underline{Q}$ .*

The main idea used behind the proof is exactly the same as the one used in CONMAX, namely the use of homotheties.

Let's formulate TOPOLMAX as a linear program. This linear program is exact, but unfortunately has infinitely many constraints. The following simple lemma forms the basis of this linear program.

**Lemma 5** *For a vector  $v \in \mathbb{R}^2$ , the inequality  $|v|_2 \leq d$  holds if and only if for each unit vector  $u \in S^1$  ( $S^1$  is the unit circle), the inequality  $u \cdot v \leq d$  holds.*

Now let's formulate our problem as a non-linear program, and then convert it to a linear program. We can define the variables  $x_1, \dots, x_m$  and  $y_1, \dots, y_m$  to be the final coordinates of the points; i.e.  $p_i^* = (x_i, y_i)$ . Our problem can be formulated like the following

$$\begin{aligned} & \text{Minimize} && s \\ & \text{Subject To} && |p_i - p_i^*|_2 \leq s \quad \forall i \in \{1, \dots, m\} \\ & && |p_i^* - p_j^*|_2 \leq 1 \quad \forall \{i, j\} \in E \end{aligned}$$

Note that this formulation can be completely written in terms of  $x_1, \dots, x_m$  and  $y_1, \dots, y_m$ ; we can simply replace each  $p_i^*$  by  $(x_i, y_i)$ . Now applying the previous lemma to this formulation, we can rewrite it like the following

$$\begin{aligned} & \text{Minimize} && s \\ & \text{Subject To} && (p_i^* - p_i) \cdot u \leq s \quad \forall i \in \{1, \dots, m\}, u \in S^1 \\ & && (p_i^* - p_j^*) \cdot u \leq 1 \quad \forall \{i, j\} \in E, u \in S^1 \end{aligned}$$

The new formulation is a linear program (although, with infinitely many constraints), since inner product is a bilinear operator. To use the ELLIPSOID method on this new formulation, we should first remove  $s$ . For

each  $s \in \mathbb{R}^{\geq 0}$ , define  $L_s$  to be the convex set in  $\mathbb{R}^{2m}$  defined by the constraints

$$\begin{aligned} (p_i^* - p_i) \cdot u &\leq s \quad \forall i \in \{1, \dots, m\}, u \in S^1 \\ (p_i^* - p_j^*) \cdot u &\leq 1 \quad \forall \{i, j\} \in E, u \in S^1 \end{aligned}$$

$L_s$  is the intersection of infinitely many half-planes. Hence, it is convex. The optimum solution of TOPOLMAX is the minimum  $s$  for which  $L_s$  is nonempty.

Because of the constraints  $(p_i^* - p_i) \cdot u \leq s$ , we can find a sphere surrounding  $L_s$ . This sphere is centered at the point  $(p_1, \dots, p_m) \in \mathbb{R}^{2m}$ , and its radius is  $\sqrt{ms}$ . That is because

$$\begin{aligned} |(p_1^*, \dots, p_m^*) - (p_1, \dots, p_m)|_2 &= \\ \sqrt{\sum_{i \in \{1, \dots, m\}} |p_i^* - p_i|_2^2} &\leq \sqrt{ms^2} = \sqrt{ms} \end{aligned}$$

Since this sphere can be surrounded by a hypercube with a side length of  $2\sqrt{ms}$ , the volume of this sphere is at most  $(2\sqrt{ms})^{2m}$ .

Note that using the previous lemma, existence of a separation oracle for  $L_s$  becomes obvious. In fact, we just have to check the unit vectors  $u$  which are parallel to the vectors  $(p_i^* - p_i)$  and the vectors  $(p_i^* - p_j^*)$ .

We have all of the things we need for the ELLIPSOID method, except  $\underline{V}$ , the lower-bound on the volume of  $L_s$ . Note that  $L_s \subseteq L_t$  for  $s \leq t$ . So if we obtain a lower-bound on the volume of  $L_s$  for one  $s$ , that lower-bound also works for every  $L_t$  for which  $t \geq s$ . Let  $OPT$  denote the optimum solution of TOPOLMAX. Let  $s^* = (1 + \delta)OPT$ . Our goal is to derive a lower-bound on the volume of  $L_{s^*}$ .

**Lemma 6** *The volume of  $L_{s^*}$  is greater than or equal to  $(\frac{\delta OPT}{2m})^{2m}$ .*

**Proof.** Since  $L_{OPT}$  is nonempty, one can find a point  $(q_1, \dots, q_m) \in L_{OPT} \subseteq L_{s^*} \subset \mathbb{R}^{2m}$ .

Let  $H^\alpha$  denote the  $\alpha$ -homothety with respect to  $q_1$ .

Consider the points  $H^\alpha(q_1), \dots, H^\alpha(q_m)$ . Since each  $q_i$  can be reached from  $q_1$  by a path consisting only of the edges in  $E$ , we have  $|q_i - q_1|_2 \leq m - 1$ . So

$$|H^\alpha(q_i) - q_1|_2 = (1 - \alpha)|q_i - q_1|_2 \leq (1 - \alpha)(m - 1)$$

Since  $|q_i - p_i| \leq OPT$ , we have  $|H^\alpha(q_i) - p_i| \leq OPT + (1 - \alpha)(m - 1)$ .

Because of the properties of homotheties, for each  $\{i, j\} \in E$ , we have  $|H^\alpha(q_i) - H^\alpha(q_j)|_2 \leq \alpha$ . Now let  $r_1, \dots, r_m$  be some arbitrary points for which we have  $|r_i - H^\alpha(q_i)|_2 \leq (1 - \alpha)/2$ . For each  $\{i, j\} \in E$ , we have

$$\begin{aligned} |r_i - r_j|_2 &\leq |H^\alpha(q_i) - H^\alpha(q_j)|_2 + |r_i - H^\alpha(q_i)|_2 \\ &\quad + |r_j - H^\alpha(q_j)|_2 \\ &\leq \alpha + \frac{1 - \alpha}{2} + \frac{1 - \alpha}{2} = 1 \end{aligned}$$

We also have

$$\begin{aligned} |r_i - p_i| &\leq |r_i - H^\alpha(q_i)| + |H^\alpha(q_i) - p_i| \\ &\leq \frac{1 - \alpha}{2} + OPT + (1 - \alpha)(m - 1) \\ &\leq OPT + (1 - \alpha)m \end{aligned}$$

This shows that there is a copy of  $\underbrace{B_{(1-\alpha)/2} \times \dots \times B_{(1-\alpha)/2}}_m$  inside  $L_{OPT+(1-\alpha)m}$ , where

$B_x$  shows a 2-dimensional ball of radius  $x$ . Since  $\text{Vol}(B_x) = \pi x^2 \geq x^2$ , we have

$$\text{Vol}(L_{OPT+(1-\alpha)m}) \geq \left(\frac{1-\alpha}{2}\right)^{2m}$$

We want  $\alpha$  to be chosen in such a way that  $OPT + (1 - \alpha)m \leq s^*$ . This can be obtained by setting  $\alpha = 1 - (s^* - OPT)/m$ . For this  $\alpha$ , we have  $1 - \alpha = (s^* - OPT)/m = \delta OPT/m$ . Therefore

$$\text{Vol}(L_{s^*}) \geq \left(\frac{\delta OPT}{2m}\right)^{2m}$$

□

Using the lower-bound  $(\frac{\delta OPT}{2m})^{2m}$  as the parameter  $\underline{V}$  of ELLIPSOID, one can see that the ellipsoid method is able to find a point inside  $L_{s^*}$  in time bounded by a polynomial of  $m$  and

$$\begin{aligned} \log \frac{(2\sqrt{m}(1 + \delta)OPT)^{2m}}{(\frac{\delta OPT}{2m})^{2m}} &= \log(4m\sqrt{m}\frac{1 + \delta}{\delta})^{2m} \\ &= 2m \log(4m\sqrt{m}\frac{1 + \delta}{\delta}) = \mathcal{O}(\text{poly}(m, 1/\delta)) \end{aligned}$$

We don't know  $OPT$ , so we can't actually set the parameter  $\underline{V}$  of ELLIPSOID to the above lower bound; this is not a problem, as we can just run the ELLIPSOID method for the time bound we have obtained (which depends only on  $m$  and  $\delta$ ).

Using the previous lemmas it's easy to see that Algorithm 3.1 is a  $(1 + \epsilon)$ -approximation for TOPOLMAX. If  $OPT$  resides in an interval  $[(1 + \delta)^i \underline{Q}, (1 + \delta)^{i+1} \underline{Q}]$ , then  $\bar{s} = (1 + \delta)^{i+2} \underline{Q}$  is definitely larger than  $s^* = (1 + \delta) \underline{Q}$ . Hence, ELLIPSOID finds a point of  $L_{\bar{s}}$  in the time limit given. But we have the following inequality (we're assuming without loss of generality that  $\epsilon \leq 1$ )

$$\begin{aligned} \bar{s} &\leq (1 + \delta)^2 (1 + \delta)^i \underline{Q} \leq (1 + \delta)^2 OPT \\ &= (1 + 2\delta + \delta^2) OPT \\ &\leq (1 + 2\delta + \delta) OPT = (1 + \epsilon) OPT \end{aligned}$$

So the solution found by Algorithm 3.1 is a  $(1 + \epsilon)$ -approximation.

**Algorithm 1** TOPOLMAX

- 
- 1: Calculate  $\underline{Q}$  using the formula  $\max\{|p_i - p_j|_2 - 1)/2 \mid \{i, j\} \in E\}$ .
  - 2: Let  $\delta = \epsilon/3$ . Divide the interval  $[\underline{Q}, 2(m-1)\underline{Q}]$  into  $\mathcal{O}(\log m / \log(1 + \epsilon))$  intervals of the form  $[(1 + \delta)^i \underline{Q}, (1 + \delta)^{i+1} \underline{Q}]$ . Sort the interval endpoints in an increasing order.
  - 3: **for** each interval endpoint like  $a$  **do**
  - 4:   Run the ELLIPSOID method on  $L_a$  using the initial bounding sphere of radius  $\sqrt{ma}$  around the origin. Run this method until it finds an answer or the upper-bound on the execution time we found earlier passes.
  - 5:   If the ELLIPSOID method finds a solution point, then stop the algorithm and return that solution.
  - 6: **end for**
  - 7: If no solution is found, return the solution found from our previous  $\mathcal{O}(m)$ -approximation algorithm.
- 

**3.2** TOPOLSUM

The same method used in the previous section can be slightly modified to work for TOPOLSUM. One can again find similar bounds on the volume of the convex body and again show that the ELLIPSOID method works in polynomial time.

**4** Concluding Remarks

In this paper we showed that FPTAS exists once the target UDG is known, i.e. adjacent vertices are specified. Therefore the hardness of CONMAX, CONSUM and similar movement problems lie in finding the topology of the target UDG. We know some good heuristic ways of guessing the target topology but have little theoretical justification for their behavior.

Considering other types of properties such as obtaining an independent set of a given size or considering a bigger class of graphs like disc graphs are good research directions to follow. Directly related to our work one can narrow the hardness and approximability gap by improving one or both. We conjecture that both CONMAX and CONSUM are approximable within constant factors.

**References**

- [1] E. Demaine, M. Hajiaghayi, H. Mahini, S. Oveisgharan, A. Sayedi, and M. Zadimoghaddam. Minimizing movement, In Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms, 2007.
- [2] E. Demaine, M. Hajiaghayi, H. Mahini, S. Oveisgharan, A. Sayedi, and M. Zadimoghaddam. Minimizing Movement, ACM Transactions on Algorithms, volume 5, number 3, July 2009, Article 30.
- [3] E. Demaine, M. Hajiaghayi, D. Marx, Minimizing Movement: Fixed-Parameter Tractability, In Proceedings of the 17th Annual European Symposium on Algorithms, 2009.
- [4] MA. Fazli, Movement Minimization in Euclidean Plane, BSc Thesis, Sharif University of Technology, 2009.
- [5] N. Ahmadipour, Movement Minimization for Network Design in Geometric Spaces, BSc Thesis, Sharif University of Technology, 2010
- [6] L.G. Valiant, University considerations in VLSI circuits, IEEE Trans. Computers 30 (1981) 135-140.
- [7] M.R. Garey, D.S. Johnson and R.E. Tarjan, The Planar Hamiltonian Problems is NP-complete, SIAM Journal on Computing, Vol. 5(1976), pp 704-714.
- [8] A. Itai, C.H. Papadimitriou and J.L. Szwarcfiter, Hamilton paths in grid graphs, SIAM Journal of Computing. 11 (1982) 676-686.
- [9] M. Mahdian, Y. Ye and J. Zhang, Approximation algorithms for metric facility location problems, SIAM Journal on Computing, 2006, p411-432.