On the Displacement for Covering a $d-$dimensional Cube with Randomly Placed Sensors[☆]

Rafał Kapelko[a,1,*], Evangelos Kranakis[b,2]

[a]Department of Computer Science,Faculty of Fundamental Problems of Technology, Wrocław University of Technology, Poland
[b]School of Computer Science, Carleton University, Ottawa, ON, Canada

Abstract

Consider n sensors placed randomly and independently with the uniform distribution in a $d-$dimensional unit cube ($d \geq 2$). The sensors have identical sensing range equal to r, for some $r > 0$. We are interested in moving the sensors from their initial positions to new positions so as to ensure that the $d-$dimensional unit cube is completely covered, i.e., every point in the $d-$dimensional cube is within the range of a sensor. If the i-th sensor is displaced a distance d_i, what is a displacement of minimum cost that ensures coverage? As cost measure for the displacement of the team of sensors we consider the *a-total movement* defined as the sum $M_a := \sum_{i=1}^{n} d_i^a$, for some constant $a > 0$. We assume that r and n are chosen so as to allow full coverage of the $d-$dimensional unit cube. Motivation for using this cost metric arises from the fact that there might be a terrain affecting the movement of the sensors from their initial to their final destinations (e.g., a terrain surface which is either obstructing or speeding the movement). Therefore the a-total displacement is not more general but can also be a more realistic metric than the one previously considered for $a = 1$.

The main contribution of this paper is to show the existence of a tradeoff in the $d-$dimensional cube between sensing radius and a-total movement. Omitting low order terms, the main results can be summarized as follows for the case of the $d-$dimensional cube.

1. If the $d-$dimensional cube sensing radius is $\frac{1}{2n^{1/d}}$ and $n = m^d$, for some $m \in \mathbb{N}$, then we present an algorithm that uses $O\left(n^{1-\frac{a}{2d}}\right)$ total expected movement (see Algorithm 2 and Theorem 5).
2. If the the $d-$dimensional cube sensing radius is greater than $\frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)}\frac{1}{2n^{1/d}}$ and n is a natural number then the total expected movement is $O\left(n^{1-\frac{a}{2d}}\left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$ (see Algorithm 3 and Theorem 7).

This sharp decline from $O(n^{1-\frac{a}{2d}})$ to $O\left(n^{1-\frac{a}{2d}}\left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$ in the a-total movement of the sensors to attain complete coverage of the $d-$dimensional cube indicates the presence of an interesting threshold on the sensing radius in a $d-$dimensional cube as it increases from $\frac{1}{2n^{1/d}}$ to $\frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)}\frac{1}{2n^{1/d}}$. In addition, we simulate Algorithm 2 and discuss the results of our simulations.

Keywords: Displacement, Random, Sensors, $d-$dimensional Cube

1. Introduction

A key challenge in utilizing effectively a group of sensors is to make them form an interconnected structure with good communication characteristics. For example, one may want to establish a sensing and communication infrastructure for robust connectivity, surveillance, security, or even reconnaissance of an urban environment using a limited number of sensors. For a team of sensors initially placed in a geometric domain such a robust connectivity cannot be assured a priori e.g., due to geographic obstacles (inhibiting transmissions), harsh environmental conditions (affecting signals), sensor faults (due to misplacement), etc. In those cases it may be required that a group of sensors originally placed in a domain be displaced to new positions either by a centralized or distributed controller. The main question arising is what is the cost of displacement so as to move the sensors from their original positions to new positions so as to attain the desired communication characteristics?

A typical sensor is able to sense a limited region usually defined by its sensing radius, say r, and considered to be a circular domain (disc of radius r). To protect a larger region against intruders every point of the region must be within the sensing range of at least one of the sensors in the group. Moreover, by forming a communication network with these sensors one is able to transmit to the entire region any disturbance that may have occurred in any part of the region. This approach has been

[☆] This is an expanded and revised version of a paper that appeared in [11] and received the Best Paper Award at the 14th International Conference, ADHOC-NOW 2015.

[*] Corresponding author at: Department of Computer Science, Faculty of Fundamental Problems of Technology, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland. Tel.: +48 71 320 33 62; fax: +48 71 320 07 51.

Email addresses: rafal.kapelko@pwr.edu.pl (Rafał Kapelko), kranakis@scs.carleton.ca (Evangelos Kranakis)

[1] Research supported by grant nr S50129/K1102

[2] Research supported in part by NSERC Discovery grant.

previously studied in several papers. It includes research on 1) *area coverage* in which one ensures monitoring of an entire region [10, 15], and 2) on *perimeter* or *barrier coverage* whereby a region is protected by monitoring its perimeter thus sensing intrusions or withdrawals to/from the interior [2, 3, 5, 6, 14]. Note that barrier coverage is less expensive (in terms of number of sensors) than area coverage. Nevertheless, barrier coverage can be only used to monitor intruders to the area, as opposed to area coverage that can also protect the interior.

1.1. Related work

Assume that n sensors of identical range are all initially placed on a line. It was shown in [5] that there is an $O(n^2)$ algorithm for minimizing the max displacement of a sensor while the optimization problem becomes NP-complete if there are two separate (non-overlapping) barriers on the line (cf. also [4] for arbitrary sensor ranges). If the optimization cost is the sum of displacements then [6] shows that the problem is NP-complete when arbitrary sensor ranges are allowed, while an $O(n^2)$ algorithm is given when all sensing ranges are the same. Similarly, if one is interested in the number of sensors moved then the coverage problem is NP-complete when arbitrary sensor ranges are allowed, and an $O(n^2)$ algorithm is given when all sensing ranges are the same [16]. Further, [7] considers the algorithmic complexity of several natural generalizations of the barrier coverage problem with sensors of arbitrary ranges, including when the initial positions of sensors are arbitrary points in the two-dimensional plane, as well as multiple barriers that are parallel or perpendicular to each other.

An important setting in considerations for barrier coverage is when the sensors are placed at random on the barrier according to the uniform distribution. Clearly, when the sensor dispersal on the barrier is random then coverage depends on the sensor density and some authors have proposed using several rounds of random dispersal for complete barrier coverage [9, 18]. Another approach is to have the sensors relocate from their initial position to a new position on the barrier so as to achieve complete coverage [5, 6, 8, 16]. Further, this relocation may be done in a centralized (cf. [5, 6]) or distributed manner (cf. [8]).

Closely related to our work is [13], where algorithm $MV_1(n,y)$ (see Algorithm 1) was analysed. In this paper, n sensors were placed in the unit interval uniformly and independently at random and the cost of displacement was measured by the sum of the respective displacements of the individual sensors in the unit line segment $[0,1]$. Lets call the positions $\frac{i}{n} - \frac{1}{2n}$, for $i = 1, 2, \ldots, n$, *anchor* positions. The sensors have the sensing radius $r = \frac{1}{2n}$ each. Notice that the only way to attain complete coverage is for the sensors to occupy the anchor positions. The following result was proved in [13].

Theorem 1 (cf. [13])**.** *Assume that, n mobile sensors are thrown uniformly and independently at random in the unit interval. The expected sum of displacements of all n sensors to move from their current location to the equidistant anchor locations* $\frac{i}{n} - \frac{1}{2n}$*, for* $i = 1, 2, \ldots, n$*, respectively, is in* $\Theta(\sqrt{n})$.

In [12], Theorem 1 was extended to when the cost of displacement is measured by the sum of the respective displacements raised to the power $a > 0$ of the respective sensors in the unit line segment $[0,1]$. The following result was proved.

Algorithm 1 $MV_1(n,y)$ (Sensor displacement on a interval).

```
Require: $n$ mobile sensors with identical sensing radius $r = \frac{y}{2n}$ placed uniformly and independently at random on the interval $[0,y]$.
Ensure: The final positions of the sensors are at the locations $\left(\frac{yi}{n} - \frac{y}{2n}\right)$, $1 \le i \le n$ (so as to attain coverage of the interval $[0,y]$.)
sort the initial locations of sensors; the locations after sorting $x_1, x_2, \ldots x_n$, $x_1 \le x_2 \le \cdots \le x_n$.
for $i = 1$ to $n$ do
    move the sensor $S_i$ at position $\left(\frac{yi}{n} - \frac{y}{2n}\right)$
end for
```

Theorem 2 (cf. [12])**.** *Assume that n mobile sensors are thrown uniformly and independently at random in the unit interval. The expected sum of displacements to a given power a of algorithm* $MV_1(n,1)$ *is in* $\Theta\left(1/n^{\frac{a}{2}-1}\right)$*, when a is natural number, and in* $O\left(1/n^{\frac{a}{2}-1}\right)$*, when* $a > 0$.

An analysis similar to the one for the line segment was provided for the unit square in [13]. Our present paper focuses on the analysis of sensor displacement for a group of sensors placed uniformly at random on the $d-$dimensional unit cube, thus also generalizing the results of [11] from $d = 2$ to arbitrary dimension $d \ge 2$. In particular, our approach is first to generalize the results of [13] to the $d-$dimensional unit cube using as cost metric the a-total movement, and second obtain sharper bounds for the case of the unit square.

1.2. Preliminaries and notation

Let d be a natural number. We define below the concept *d-Dimensional Cube Sensing Radius* which refers to a coverage area having the shape of a d-dimensional cube.[3]

Definition 3 (d-Dimensional Cube Sensing Radius)**.** *Consider a sensor* $Z_{(x_1,x_2,\ldots,x_d)}$ *located in position* $(x_1, x_2, \ldots, x_d)$*, where* $0 \le x_1, x_2, \ldots, x_d \le 1$. *We define the range of the sensor* $Z_{(x_1,x_2,\ldots,x_d)}$ *to be the area delimited by the d-dimensional cube with the* 2^d *vertices* $(x_1 \pm r, x_2 \pm r, \ldots x_d \pm r)$*, and call r the d-dimensional cube sensing radius of the sensor.*

We also define the cost measure *a-total movement* as follows.

Definition 4 (a-total movement)**.** *Let* $a > 0$ *be a constant. Suppose the displacement of the i-th sensor is a distance* d_i*. The* a-total movement *is defined as the sum* $M_a := \sum_{i=1}^{n} d_i^a$*. (We assume that, r and n are chosen so as to allow full coverage of the d-dimensional cube and* $a > 0$*.)*

[3] Recall that the generally accepted coverage area of a sensor is a d-dimensional ball. Our results can be easily converted to this model by d-dimensional ball circumscribing the d-dimensional cube.

In the analysis below we consider the Beta distribution. We say that a random variable concentrated on the interval $[0, 1]$ has the $\mathrm{B}(a, b)$ distribution with parameters a, b, if it has the probability density function

$$f(x) = \frac{1}{\mathrm{B}(a, b)} x^{a-1}(1 - x)^{b-1}, \tag{1}$$

where the Euler Beta function (see [17])

$$\mathrm{B}(a, b) = \int_0^1 x^{a-1}(1 - x)^{b-1} dx \tag{2}$$

is defined for all complex numbers a, b such as $\Re(a) > 0$ and $\Re(b) > 0$. Let us notice that for any integer numbers $a, b \geq 0$, we have

$$\mathrm{B}(a, b)^{-1} = \binom{a + b - 1}{a} a. \tag{3}$$

1.3. Results of the paper

We consider n mobile sensors with identical $d-$dimensional cube sensing radius r placed independently at random with the uniform distribution in the $d-$dimensional unit cube ($d \geq 2$). We want to have the sensors move from their current location to positions that cover the $d-$dimensional cube in the sense that every point in the $d-$dimensional cube is within the range of at least one sensor. When a sensor is displaced on the $d-$dimensional cube a distance equal to d_x the cost of the displacement is d_x^a for some (fixed) power $a > 0$ of the distance d_x traveled. We assume that r and n are chosen so as to allow full coverage of the $d-$dimensional cube, i.e., every point of the region is within the range of at least one sensor.

Omitting unnecessary low order terms, the main contribution of the paper in Section 2 is to show the existence of a tradeoff between $d-$dimensional cube sensing radius and a-total movement that can be summarized as follows:

1. For the case of the $d-$dimensional cube sensing radius $\frac{1}{2n^{1/d}}$ and $n = m^d$ for some $m \in \mathbb{N}$ we present an algorithm that uses $O\left(n^{1-\frac{a}{2d}}\right)$ total expected movement (see Algorithm 2 and Theorem 5).
2. If the $d-$dimensional cube sensing radius is greater than $\frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)} \frac{1}{2n^{1/d}}$ and n is a natural number then the expected movement is $O\left(n^{1-\frac{a}{2d}} \left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$ (see Algorithm 3 and Theorem 7).

Notice that, for $a = d$ Algorithm $MV_d(n, 1)$ (see Algorithm 2) uses $O(\sqrt{n})$ total expected movement while Algorithm $LV_d(n)$ (see Algorithm 3) uses $O(\sqrt{\ln n})$ total expected movement. Therefore this sharp decrease from $O(n^{1-\frac{a}{2d}})$ to $O\left(n^{1-\frac{a}{2d}} \left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$ in the a-total movement of the sensors to attain complete coverage of the $d-$dimensional cube indicates the presence of an interesting threshold on the $d-$dimensional cube sensing radius when it increases from $\frac{1}{2n^{1/d}}$ to $\frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)} \frac{1}{2n^{1/d}}$.

In Section 3 we simulate Algorithm 2 and provide the results of the simulations. Finally, Section 4 is the conclusion.

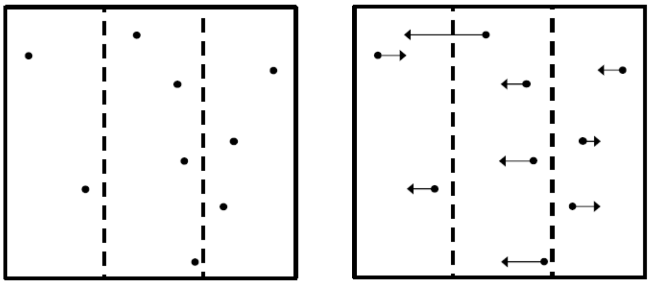

Fig. 1: Nine mobile sensors located in the interrior of a unit square move to new positions according to steps $(1 - 6)$ of Algorithm $MV_2(9, 1)$.

2. Displacement in $d-$dimensional cube

Assume that n mobile sensors with the same $d-$dimensional cube sensing radius are thrown uniformly and independently at random in the $d-$dimensional unit cube $[0, 1]^d$. Let $a > 0$ and $d \geq 2$.

Our first result is an upper bound on the expected $a-$total movement for the case, where the $d-$dimensional cube sensing radius is $\frac{1}{2n^{1/d}}$. Observe that in this case the only way for the sensors to attain complete coverage of the $d-$dimensional unit cube is to occupy the positions

$$\left(\frac{l_1}{n^{1/d}} - \frac{1}{2n^{1/d}}, \frac{l_2}{n^{1/d}} - \frac{1}{2n^{1/d}}, \dots, \frac{l_d}{n^{1/d}} - \frac{1}{2n^{1/d}}\right),$$

for $1 \leq l_1, l_2, \dots, l_d \leq n^{1/d}$ and $l_1, l_2, \dots, l_d \in \mathbb{N}$. We present a recursive algorithm $MV_d(n, 1)$ (see Algoritm 2 for $y = 1$) that uses $O\left(n^{1-\frac{a}{2d}}\right)$ expected $a-$total movement.

The phases of the algorithm are d. When the sensors are in the $i-$th phase then they are moved in the $(d-i+1)-$dimensional cubes. During the first phase (see steps $(1 - 6)$) we apply a greedy strategy and move all the sensors only according to the first coordinate. Figure 1 illustrates the steps $(1 - 6)$ of Algorithm $MV_2(n, 1)$ As a result of the first phase we have $n^{1/d}$ $(d - 1)$-dimensional cubes each with $n^{\frac{d-1}{d}}$ random sensors. Therefore, $n = m^d$ for some $m \in \mathbb{N}$. Hence the first phase reduces the sensor movement on the unit d-dimensional cube to the sensor movement on the unit $(d - 1)$-dimensional cube. Then (see steps $(7 - 9)$) we move sensors in the unit $(d - 1)$-dimensional cube. Notice that for the base case $d = 1$ we execute algorithm $MV_1(n, 1)$.

We prove the following theorem.

Theorem 5. *Fix $d \in \mathbb{N}$. Let $n = m^d$ for some $m \in \mathbb{N}$ and let $a > 0$. Assume that n sensors of d-dimensional cube sensing radius equal to $\frac{1}{2n^{1/d}}$ are thrown randomly and uniformly and independently with the uniform distribution on a unit d-dimensional cube. The expected $a-$total movement of algorithm $MV_d(n, 1)$ is in $O\left(n^{1-\frac{a}{2d}}\right)$.*

Proof. First of all, observe that $n = m^d$ for some $m \in \mathbb{N}$ (see step (8) of Algorithm 2). We will prove the statement of the theorem by mathematical induction. Observe that the base case for $d = 1$ follows from Theorem 2 [cf. [12]]. Let us assume the

Algorithm 2 $MV_d(n,y)$ Sensor displacement on a d-dimensional cube $[0,y]^d$, when n is dth power of the natural number and $d \geq 2$.

```
Require: $n$ mobile sensors with identical $d$-dimensional cube sensing radius $r = \frac{1}{2n^{1/d}}$ placed uniformly and independently at random on the $d-$dimensional cube $[0,y]^d$
Ensure: The final positions of the sensors are at the locations $\left(\frac{yl_1}{n^{1/d}} - \frac{y}{2n^{1/d}}, \frac{yl_2}{n^{1/d}} - \frac{yl_2}{2n^{1/d}}, \dots, \frac{yl_d}{n^{1/d}} - \frac{yl_d}{2n^{1/d}}\right)$, $1 \leq l_1, l_2, \dots, l_d \leq n^{1/d}$
    and $l_1, l_2, \dots, l_d \in \mathbb{N}$ (so as to attain coverage of $[0,y]^d)$
1: Sort the initial locations of sensors according to the first coordinate; the locations
   after sorting $S_1 = (x_1(1), x_2(1), \dots, x_d(1)), S_2 = (x_1(2), x_2(2), \dots, x_d(2)), \dots$
   $S_n = (x_1(n), x_2(n), \dots, x_d(n))$, $x_1(1) \leq x_1(2) \leq \dots \leq x_1(n)$;
2: for $j = 1$ to $n^{1/d}$ do
3:     for $i = 1$ to $n^{(d-1)/d}$ do
4:         Move the sensor $S_{i+(j-1)n^{1/d}}$ at position
           $\left(\frac{jy}{n^{1/d}} - \frac{y}{2n^{1/d}}, x_{i+(j-1)n^{1/d}}(2), \dots, x_{i+(j-1)n^{1/d}}(d)\right)$;
5:     end for
6: end for
7: for $j = 1$ to $n^{1/d}$ do
8:     Execute $MV_{d-1}(n^{(d-1)/d}, y)$ for sensors
       $S_{1+(j-1)n^{1/d}}, S_{2+(j-1)n^{1/d}}, \dots, S_{n^{(d-1)/d}+(j-1)n^{1/d}}$;
9: end for
```

result holds for the number $d-1$. Let $a > 0$. We will estimate the expected $a-$total movement at the steps $(1-6)$. Let X_i be the ith order statistic, i.e., the position of the ith sensor in the interval $[0,1]$ after sorting in step (1). It turns out (see [1]) that X_i obeys the Beta distribution with parameters $i, n-i+1$. We know that the density function for X_i (see Equation (1)) is

$$f_{X_i}(x) = i\binom{n}{i} x^{i-1}(1-x)^{n-i}.$$

Therefore, the expected $a-$total movement in steps $(1-6)$ of algorithm $MV_d(n,1)$ is equal to

$$E^{(a)}_{(1-6)} = \sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{(d-1)/d}+1}^{jn^{(d-1)/d}} i\binom{n}{i} \int_0^1 \left| x - \left(\frac{j}{n^{1/d}} - \frac{1}{2n^{1/d}}\right)\right|^a \times$$
$$\times\, x^{i-1}(1-x)^{n-i}dx.$$

Notice that, the expected $a-$total movement of algorithm $MV_1(n,1)$ is equal to

$$\begin{aligned} D^{(a)} &= \sum_{i=1}^{n} i\binom{n}{i} \int_0^1 \left| x - \left(\frac{i}{n} - \frac{1}{2n}\right)\right|^a x^{i-1}(1-x)^{n-i}dx \\ &= \sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{(d-1)/d}+1}^{jn^{(d-1)/d}} i\binom{n}{i} \int_0^1 \left| x - \left(\frac{i}{n} - \frac{1}{2n}\right)\right|^a \times \\ &\times\, x^{i-1}(1-x)^{n-i}dx. \end{aligned}$$

According to Theorem 2 [cf. [12]]

$$D^{(a)} = O(n^{1-\frac{a}{2}}), \tag{4}$$

when $a > 0$.

Firstly, we estimate $E^{(a)}_{(1-6)}$, when $a \geq 1$. Notice that

$$|z_1 + z_2|^a \leq (|z_1| + |z_2|)^a \leq 2^{a-1}(|z_1|^a + |z_2|^a) \text{ for } a \geq 1, z_1, z_2 \in \mathbb{R}. \tag{5}$$

This inequality is the consequence of the fact that $f(x) = x^a$ is convex over $\mathbb{R}_+$ for $a \geq 1$. Using Inequality (5) for $z_1 = x - \left(\frac{i}{n} - \frac{1}{2n}\right)$ and $z_2 = \left(\frac{i}{n} - \frac{1}{2n}\right) - \left(\frac{j}{n^{1/d}} - \frac{1}{2n^{1/d}}\right)$ we get

$$\begin{aligned} &\left| x - \left(\frac{j}{n^{1/d}} - \frac{1}{2n^{1/d}}\right)\right|^a \\ &\leq 2^{a-1}\left(\left| x - \left(\frac{i}{n} - \frac{1}{2n}\right)\right|^a + \left|\frac{i}{n} - \frac{j}{n^{1/d}} + \frac{1}{2n^{1/d}} - \frac{1}{2n}\right|^a\right). \end{aligned} \tag{6}$$

We apply the definition of the Beta function (see Equation (2)) with parameters $i, n-i+1$, as well as Equation (3) to deduce that

$$\begin{aligned} &\sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{(d-1)/d}+1}^{jn^{(d-1)/d}} i\binom{n}{i} \int_0^1 \left|\frac{i}{n} - \frac{j}{n^{1/d}} + \frac{1}{2n^{1/d}} - \frac{1}{2n}\right|^a \times \\ &\quad \times\, x^{i-1}(1-x)^{n-i}dx \\ &\quad = \sum_{j=1}^{n^{1/d}} \sum_{k=1}^{n^{(d-1)/d}} \left|\frac{(j-1)n^{(d-1)/d} + k}{n} - \frac{j}{n^{1/d}} + \frac{1}{2n^{1/d}} - \frac{1}{2n}\right|^a \\ &\quad = \sum_{j=1}^{n^{1/d}} \sum_{k=1}^{n^{(d-1)/d}} \left|\frac{k}{n} - \frac{1}{2n^{1/d}} - \frac{1}{2n}\right|^a \\ &\quad \leq \sum_{j=1}^{n^{1/d}} \sum_{k=1}^{n^{(d-1)/d}} \left(\frac{1}{2n^{1/d}}\right)^a = \frac{n^{1-\frac{a}{d}}}{2^a}. \end{aligned} \tag{7}$$

Putting together Formulas (4), (6), and (7) we obtain

$$E^{(a)}_{(1-6)} = O(n^{1-\frac{a}{2}}) + O(n^{1-\frac{a}{d}}) = O(n^{1-\frac{a}{d}}), \text{ when } a \geq 1, d \geq 2 \tag{8}$$

To estimate $E^{(a)}_{(1-6)}$, when $0 < a < 1$ we define

$$F^{(a)}_{(i,j)} = i\binom{n}{i} \int_0^1 \left| x - \left(\frac{j}{n^{1/d}} - \frac{1}{2n^{1/d}}\right)\right|^a x^{i-1}(1-x)^{n-i}dx.$$

Observe that,

$$E^{(a)}_{(1-6)} = \sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{(d-1)/d}+1}^{jn^{(d-1)/d}} F^{(a)}_{(i,j)}.$$

Then, we use the discrete Hölder inequality with parameters $\frac{1}{a}$

and $\frac{1}{1-a}$ to derive

$$\sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{\frac{d-1}{d}}+1}^{jn^{\frac{d-1}{d}}} F^{(a)}_{(i,j)}$$

$$\leq \left(\sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{\frac{d-1}{d}}+1}^{jn^{\frac{d-1}{d}}} \left(F^{(a)}_{(i,j)}\right)^{\frac{1}{a}} \right)^{\frac{a}{1}} \left(\sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{\frac{d-1}{d}}+1}^{jn^{\frac{d-1}{d}}} 1 \right)^{1-a}$$

$$\leq \left(\sum_{j=1}^{n^{1/d}} \sum_{i=(j-1)n^{(d-1)/d}+1}^{jn^{(d-1)/d}} \left(F^{(a)}_{(i,j)}\right)^{\frac{1}{a}} \right)^{a} (n)^{1-a}. \tag{9}$$

Next, we use Hölder inequality for integrals with parameters $\frac{1}{a}$ and $\frac{1}{1-a}$ and get

$$\int_0^1 \left| x - \left(\frac{j}{n^{1/d}} - \frac{1}{2n^{1/d}} \right) \right|^a x^{i-1}(1-x)^{n-i} i \binom{n}{i} dx$$

$$\leq \left(\int_0^1 \left(\left| x - \left(\frac{j}{n^{1/d}} - \frac{1}{2n^{1/d}} \right) \right|^a \right)^{\frac{1}{a}} x^{i-1}(1-x)^{n-i} i \binom{n}{i} dx \right)^{\frac{a}{1}},$$

so $\left(F^{(a)}_{(i,j)}\right)^{\frac{1}{a}} \leq F^{(1)}_{(i,j)}$. Putting together Equation (8) and Equation (9) we obtain

$$E^{(a)}_{(1-6)} = O\left(n^{1-\frac{a}{d}}\right), \text{ when } 0 < a < 1. \tag{10}$$

Observe that in step (8) of algorithm $MV_d(n,1)$ we have that $n^{\frac{d-1}{d}}$ mobile sensors are thrown uniformly and indendently at random in the unit $(d-1)$-dimensional cube. According to inductive assumption the expected $a-$total movement at the step (8) is equal $O((n^{\frac{d-1}{d}})^{1-\frac{a}{2(d-1)}})$. Hence the expected $a-$total movement in steps $(7-9)$ is in $O(n^{1/d}(n^{\frac{d-1}{d}})^{1-\frac{a}{2(d-1)}}) = O(n^{1-\frac{a}{2d}})$. Notice that the expected $a-$total movement in steps (1-6) is equal $O(n^{1-\frac{a}{d}})$ (see Formula (8) and Formula (10)). Therefore, the expected cost of displacement to power a of algorithm $MV_d(n,1)$ is in $O(n^{1-\frac{a}{2d}})$. This gives the claimed estimate for d and completes the proof of Theorem 5. □

Now we study an upper bound on the total displacement, when the $d-$dimensional cube sensing radius of the sensors is larger than $\frac{1}{2n^{1/d}}$. We begin with a lemma which will be helpful in the proof of Theorem 7. The lemma indicates how to scale the results of Theorem 5 to $d-$dimensional cube of arbitrary length and states that Algorithm $MV_d(n,y)$ uses $O\left(y^a n^{1-\frac{a}{2d}}\right)$ expected $a-$total movement.

Lemma 6. *Fix $d \in \mathbb{N}$. Let $n = m^d$ for some $m \in \mathbb{N}$ and let $a > 0$. Assume that n sensors of d-dimensional cube sensing radius equal to $\frac{y}{2n^{1/d}}$ are thrown randomly and uniformly and independently with the uniform distribution on the $[0,y]^d$. The expected $a-$total movement of algorithm $MV_d(n,y)$ is in $O\left(y^a n^{1-\frac{a}{2d}}\right)$.*

Proof. Assume that, n sensors are in the cube $[0,y]^d$. Then, multiply their coordinates by $1/y$. From Theorem 5 the expected $a-$total movement in the unit cube $[0,1]^d$ is in $O\left(n^{1-\frac{a}{2d}}\right)$. Now by multiplying their coordinates by y we get the result in the statement of the lemma. □

A natural question to ask is: how to exploit the proposed Algorithm $MV_d(n,1)$ when the number n of nodes is not a $d-$th power of natural number. Assume that n sensors have the $d-$dimensional cube sensing radius $r = \frac{f}{2n^{1/d}}$ and $f \geq \frac{n^{1/d}}{\lfloor n^{1/d} \rfloor}$. To attain coverage of the cube $[0,1]^d$ choose $\lfloor n^{1/d} \rfloor^d$ sensors at random and use Algorithm $MV_d(n,1)$ for the chosen sensors. Then similar arguments hold for Algorithm $MV_d(n,y)$.

Notice that for $f \geq \frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)}$ we can do better. The following theorem states that Algorithm $LV_d(n)$ uses $O\left(n^{1-\frac{a}{2d}} \left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$ expected $a-$total movement.

Algorithm 3 $LV_d(n)$ Sensor displacement on a unit d-dimensional cube when $d \geq 2$, $p = \frac{9}{4}\left(2 + \frac{a}{d}\right)$, $A = \frac{3}{4}\left(2 + \frac{a}{d}\right)$, x_0 is the real solution of the equation $\frac{x}{\frac{9}{4}(2+\frac{a}{d})\ln x} = 3$ such that $x_0 \geq 3$

```
Require: $n \geq \lceil x_0 \rceil$ mobile sensors with identical square sensing radius $r = \frac{f}{2n^{1/d}}$, $f \geq \frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)}$ placed uniformly and independently at random on the cube $[0,1]^d$.
Ensure: The final positions of sensors to attain coverage of the cube $[0,1]^d$
Divide the $d-$dimensional unit cube into $d-$dimensional subcubes of side $\frac{1}{\left\lfloor \left(\frac{n}{p \ln n}\right)^{1/d} \right\rfloor}$;
if there is a $d$-dimensional subcube with fewer than $\frac{1}{3} \frac{n}{\left\lfloor \left(\frac{n}{p \ln n}\right)^{1/d} \right\rfloor^d}$ sensors then
    choose $\lfloor n^{1/d} \rfloor^d$ sensors at random;
    use Algorithm $MV_d(n,1)$ that moves all $n := \lfloor n^{1/d} \rfloor^d$ sensors to equidistant points that are sufficient to cover the $d$-dimensional subcube;
else
    In each $d$-dimensional subcube choose $\left\lfloor (A \ln n)^{1/d} \right\rfloor^d$ sensors at random and use Algorithm $MV_d(n,y)$ with $n := \left\lfloor (A \ln n)^{1/d} \right\rfloor^d$, $y := \frac{1}{\left\lfloor \left(\frac{n}{p \ln n}\right)^{1/d} \right\rfloor}$ to move the chosen sensors to equidistant positions so as to cover the $d$-dimensional subcube;
end if
```

Theorem 7. *Fix $d \in \mathbb{N} \setminus \{1\}$ and $a > 0$. Let $f \geq \frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)}$ and $n \geq \lceil x_0 \rceil$, where x_0 is the real solution of the equation $\frac{x}{\frac{9}{4}(2+\frac{a}{d})\ln x} = 3$ such that $x_0 \geq 3$. Assume that n sensors of d-dimensional cube sensing radius $r = \frac{f}{2n^{1/d}}$ are thrown randomly and uniformly and independently with the uniform distribution on the $[0,1]^d$. The expected $a-$total movement of algorithm $LV_d(n)$ is in $O\left(n^{1-\frac{a}{2d}} \left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$.*

Proof. Assume that $d \in \mathbb{N} \setminus \{1\}$ and $a > 0$. Let $p = \frac{9}{4}\left(2 + \frac{a}{d}\right)$ and $A = \frac{3}{4}\left(2 + \frac{a}{d}\right)$, x_0 is the real solution of the equation $\frac{x}{\frac{9}{4}(2+\frac{a}{d})\ln x} = 3$ such that $x_0 \geq 3$. First of all, observe that $\frac{n}{p\ln(n)} > 3$ for $n \geq \lceil x_0 \rceil$. We will prove that Algorithm $LV_d(n)$ uses $O\left(n^{1-\frac{a}{2d}} \left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$ expected $a-$total movement. There are two cases to consider.

Case 1: There exists a $d-$dimensional subcube with fewer than

$$\frac{1}{3}\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}$$

sensors. In this case choose $\lfloor n^{1/d}\rfloor^d$ sensors uniformly and randomly from n sensors. Applying the inequalities $\lfloor x\rfloor > x-1$ and $f \geq \frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)} > \frac{3^{1/d}}{3^{1/d}-1}$ we deduce that

$$\left(\lfloor n^{1/d}\rfloor\frac{f}{n^{1/d}}\right)^d > \left(\frac{n^{1/d}-1}{n^{1/d}}\frac{3^{1/d}}{3^{1/d}-1}\right)^d \geq 1 \text{ for } n \geq 3.$$

Therefore, the $\lfloor n^{1/d}\rfloor^d$ chosen sensors are enough to attain the coverage. The expected $a-$total movement is $O\left(\left(\lfloor n^{1/d}\rfloor^d\right)^{1-\frac{a}{2d}}\right) = O\left(n^{1-\frac{a}{2d}}\right)$ by Theorem 5.

Case 2: All $d-$dimensional subcubes contain at least $\frac{1}{3}\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}$ sensors. From the inequality $\lfloor x\rfloor \leq x$ we deduce that,

$$\left\lfloor (A\ln n)^{1/d}\right\rfloor^d \leq \frac{1}{3}\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}.$$

Hence it is possible to choose $\left\lfloor (A\ln n)^{1/d}\right\rfloor^d$ sensors at random in each d-dimensional subcube with more than $\frac{1}{3}\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}$ sensors. Let us consider the sequence

$$a_n = \frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)}\left\lfloor (A\ln n)^{1/d}\right\rfloor\frac{1}{n^{1/d}}\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor$$

for $n \geq \lceil x_0\rceil$. Applying inequality $\lfloor x\rfloor > x-1$ we see that

$$\begin{aligned} a_n &> \frac{3^{3/d}}{(3^{1/d}-1)(3^{1/d}-1)}\left((A\ln n)^{1/d}-1\right)\frac{1}{n^{1/d}}\times \\ &\times\left(\left(\frac{n}{p\ln n}\right)^{1/d}-1\right) \\ &= \frac{3^{2/d}}{(3^{1/d}-1)(3^{1/d}-1)}\left(1-\frac{1}{(A\ln n)^{1/d}}\right)\times \\ &\left(1-\left(\frac{p\ln n}{n}\right)^{1/d}\right) \end{aligned} \tag{11}$$

Observe that

$$\frac{p\ln n}{n} \leq \frac{1}{3},\ \frac{1}{A\ln n} \leq \frac{1}{3} \text{ for } n \geq \lceil x_0\rceil \tag{12}$$

Putting together Equation (11) and Equation (12) we get

$$\left\lfloor (A\ln n)^{1/d}\right\rfloor^d\frac{f^d}{n}\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d \geq a_n^d > 1.$$

Therefore, $\left\lfloor (A\ln n)^{1/d}\right\rfloor^d$ chosen sensors are enough to attain the coverage. By the independence of the sensors positions, the $\left\lfloor (A\ln n)^{1/d}\right\rfloor^d$ chosen sensors in any given d-dimensional subcube are distributed randomly and independently with uniform distribution over the d-dimensional subcube of side $y = \frac{1}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor}$. By Lemma 6 the expected $a-$total movement inside each d-dimensional subcube is

$$O\left(\left(\frac{1}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor}\right)^a\left(\left\lfloor (A\ln n)^{1/d}\right\rfloor^d\right)^{1-\frac{a}{2d}}\right) = O\left(\frac{(\ln n)^{\frac{a}{2d}}}{n^{\frac{a}{d}}}(\ln n)\right).$$

Since, there are $\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d$ d-dimensional subcubes, the expected $a-$total movement over all d-dimensional subcubes must be in $O\left(n^{1-\frac{a}{2d}}\left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)$. It remains to consider the probability with which each of these cases occurs. The proof of the theorem will be a consequence of the following Claim.

Claim 8. *Let $p = \frac{9}{4}\left(2+\frac{a}{d}\right)$. The probability that fewer than $\frac{1}{3}\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}$ sensors fall in any d-dimensional subcube is <* $\frac{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}{n^{1+\frac{a}{2d}}}$.

Proof. (Claim 8) First of all, from the inequality $\lfloor x\rfloor \leq x$ we get

$$\sqrt{\frac{\left(2+\frac{a}{d}\right)\ln n}{n}\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d} \leq \frac{2}{3}.$$

Hence,

$$\frac{1}{3}\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d} \leq \frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d} - \sqrt{\frac{\left(2+\frac{a}{d}\right)n\ln n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}}. \tag{13}$$

The number of sensors falling in a d-dimensional subcube is a Bernoulli process with probability of success $\frac{1}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}$. By Chernoff bounds, the probability that a given d-dimensional subcube has fewer than

$$\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d} - \sqrt{\frac{\left(2+\frac{a}{d}\right)n\ln n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}}$$

sensors is less than $e^{-\left(1+\frac{a}{2d}\right)\ln n} < \frac{1}{n^{1+\frac{a}{2d}}}$. Specifically we use the Chernoff bound

$$\Pr[X < (1-\delta)m] < e^{-\delta^2 m/2},$$

$m = \frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}$, $\delta = \sqrt{\frac{\left(2+\frac{a}{d}\right)\ln n}{n}\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}$. As there are $\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d$ d-dimensional subcubes, the event that one has fewer than

$$\frac{n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d} - \sqrt{\frac{\left(2+\frac{a}{d}\right)n\ln n}{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}}.$$

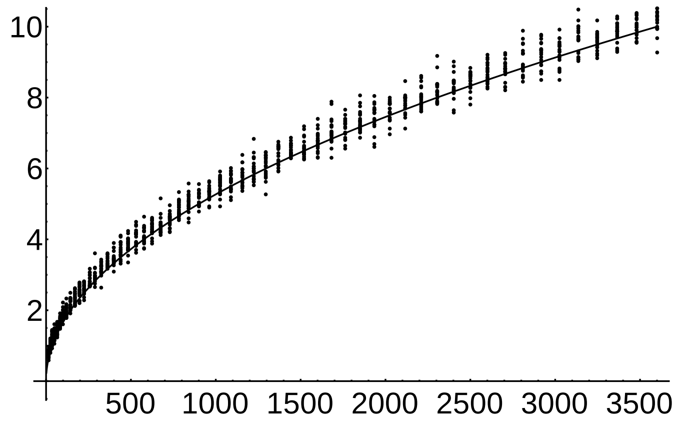

Fig. 2: The expected 2−total movement of Algorithm $MV_2(n,1)$; $M_2 = \frac{1}{6}\sqrt{n}$

sensors occurs with probability less than $\frac{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}{n^{1+\frac{a}{2d}}}$. This and Equation (13) completes the proof of Claim 8. □

Using Claim 8 we can upper bound the expected a−total movement as follows:

$$\left(1-\frac{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}{n^{1+\frac{a}{2d}}}\right)O\left(n^{1-\frac{a}{2d}}\left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right)+\left(\frac{\left\lfloor\left(\frac{n}{p\ln n}\right)^{1/d}\right\rfloor^d}{n^{1+\frac{a}{2d}}}\right)O\left(n^{1-\frac{a}{2d}}\right)$$

$$=O\left(n^{1-\frac{a}{2d}}\left(\frac{\ln n}{n}\right)^{\frac{a}{2d}}\right),$$

which proves Theorem 7. □

3. Simulation Results

In this section we use simulation results to analyze how random placement of sensors on the square impacts the expected a−total movement.

We repeated 20 times the following experiments. Firstly, for each number of sensors $n \in \{2^2, 3^2, 4^2, \ldots, 60^2\}$ we generated 10 random placements. Then we calculated the expected a−total movement according to Algorithm $MV_2(n,y)$. Let $E_{n,10}$ be the average of 10 measurements of the expected a−total movement. Then, we placed the points in the set $\{(n, E_{n,10}) : n = 2^2, 3^2, 4^2, \ldots, 60^2\}$ into the picture.

Figures 2 and 3 illustrates the described experiments for Algorithm $MV_2(n,1)$ when $a = 2$ and $a = 4$. The additional line in the above pictures is the plot of the function which is the theoretical estimation. Black dots which represent numerical results are situated near the theoretical line. According to the proof of Theorem 5 the steps (7-9) of Algorithm $MV_2(n,1)$ contribute the asymptotics. Notice that, the expected a−total movement in steps (7-9) of Algorithm $MV_2(n,1)$ is equal to

$$E^{(a)}_{(7-9)} = \sqrt{n}\sum_{i=1}^{\sqrt{n}} i\binom{\sqrt{n}}{i}\int_0^1\left|x-\left(\frac{j}{\sqrt{n}}-\frac{1}{2\sqrt{n}}\right)\right|^a \times$$
$$\times\, x^{i-1}(1-x)^{\sqrt{n}-i}dx.$$

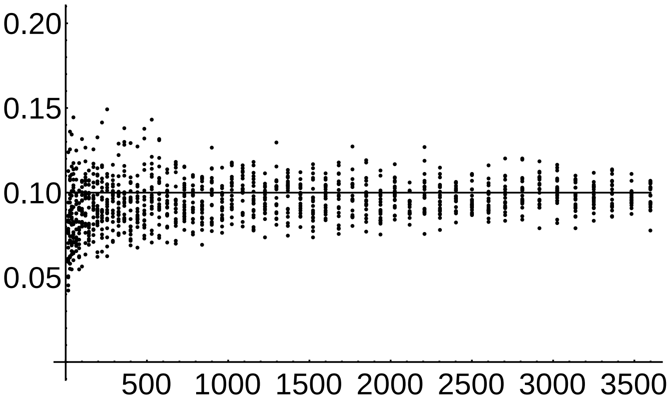

Fig. 3: The expected 4−total movement of Algorithm $MV_2(n,1)$; $M_4 = \frac{1}{10}$

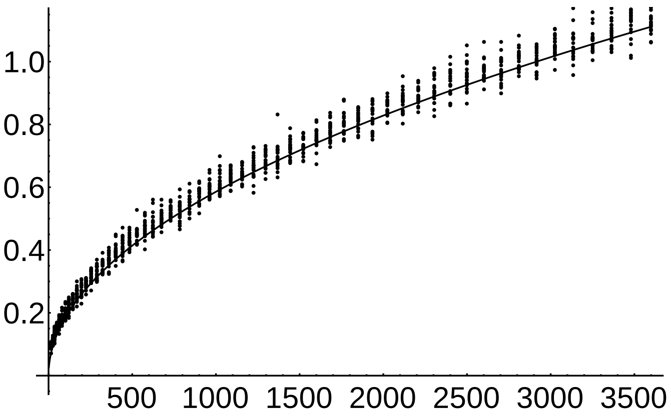

Fig. 4: The expected 2−total movement of Algorithm $MV_2\left(n,\frac{1}{3}\right)$; $M_2 = \frac{1}{54}\sqrt{n}$

Applying the Formulas for $E^{(2)}_{(7-9)}$ and $E^{(4)}_{(7-9)}$ in any mathematical software that performs symbolic calculation we get

$$E^{(2)}_{(7-9)} \sim \frac{1}{6}\sqrt{n} \text{ and } E^{(4)}_{(7-9)} \sim \frac{1}{10}.$$

Therefore, $M_2 = \frac{1}{6}\sqrt{n}$ and $M_4 = \frac{1}{10}$.

Figures 4 and 5 illustrates the described experiments for Algorithm $MV_2(n,\frac{1}{3})$ when $a = 2$ and $MV_2(n,2)$ when $a = 4$. The additional line in the above pictures is the plot of the function which is the theoretical estimation. Black dots which represent numerical results are situated near the theoretical line. According to the proof of Lemma 6 we have $M_2 = \left(\frac{1}{3}\right)^2\frac{1}{6}\sqrt{n} = \frac{1}{54}\sqrt{n}$ and $M_4 = 2^4\frac{1}{10} = \frac{4}{5}$.

3.1. Mathematica code

The following Mathematica code was used to simulate Algorithm 2 for $d = 2$, $a = 2$ and $y = 1$.

```
CH2[n_]:= Block[{M1,L1,M2,L2 = 0,L},
  M1=Sort[RandomReal[{0, 1},n]];
  L1=Sum[Sum[(M1[[i]] - j/Sqrt[n]
     +1/(2*Sqrt[n]))^2,
   {i,(j - 1)*Sqrt[n] + 1,j*Sqrt[n]}],
```

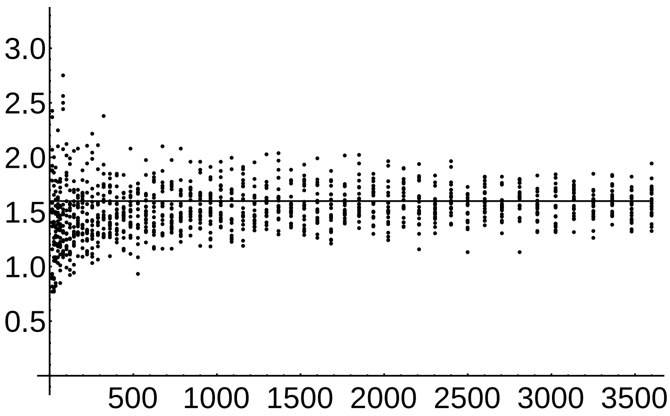

Fig. 5: The expected 4−total movement of Algorithm $MV_2(n, 2)$; $M_4 = \frac{4}{5}$

```
    {j,1,Sqrt[n]}];
  Do[M2 = Sort[RandomReal[{0,1},Sqrt[n]]];
      L2=L2+Sum[(M2[[i]]-(i/Sqrt[n]
         -1/(2*Sqrt[n])))^2,{i,1,Sqrt[n]}],
      {j,1,Sqrt[n]}];
      L=L1+L2];

SR[n_,IP_]:=Mean[Table[CH2[n],{IP}]];

points[m_]:=Block[{data={}},
 Do[data=Join[data,Table[{n^2,SR[n^2,10]},
     {n,4,60,1}]],{counter,1,m}];data];

Plot[{Sqrt[n]/6},{n,2,3600},
   PlotStyle->Directive[Thick, Black],
   Epilog->{PointSize[Medium],
   Point[points[20]]},
   TicksStyle->Directive[Black,35],
   AxesOrigin->{0,0},
   AxesStyle -> Directive[Thick]]
```

4. Conclusion

In this paper we studied the movement of n sensors with identical square sensing radius in d dimensions when the cost of movement of sensor is proportional to some (fixed) power $a > 0$ of the distance traveled. We obtained bounds on the movement depending on the range of sensors. An interesting problem for future work is to evaluate experimental results from some real-life sensor deployments. Also experimental validation of the results would be useful. However this might be rather expensive due to the large number of sensors that would be required for reliable estimation.

References

[1] B. Arnold, N. Balakrishnan, and H. Nagaraja. *A first course in order statistics*, volume 54. SIAM, 1992.

[2] P. Balister, B. Bollobas, A. Sarkar, and S. Kumar. Reliable density estimates for coverage and connectivity in thin strips of finite length. In *Proceedings of MobiCom '07*, pages 75–86. ACM, 2007.

[3] B. Bhattacharya, M. Burmester, Y. Hu, E. Kranakis, Q. Shi, and A. Wiese. Optimal movement of mobile sensors for barrier coverage of a planar region. *Theoretical Computer Science*, 410(52):5515 – 5528, 2009.

[4] D. Z. Chen, Y. Gu, J. Li, and H. Wang. Algorithms on minimizing the maximum sensor movement for barrier coverage of a linear domain. In *Proceedings of SWAT'12*, pages 177–188, 2012.

[5] J. Czyzowicz, E. Kranakis, D. Krizanc, I. Lambadaris, L. Narayanan, J. Opatrny, L. Stacho, J. Urrutia, and M. Yazdani. On minimizing the maximum sensor movement for barrier coverage of a line segment. In *Proceedings of ADHOC-NOW, LNCS v. 5793*, pages 194–212, 2009.

[6] J. Czyzowicz, E. Kranakis, D. Krizanc, I. Lambadaris, L. Narayanan, J. Opatrny, L. Stacho, J. Urrutia, and M. Yazdani. On minimizing the sum of sensor movements for barrier coverage of a line segment. In *Proceedings of ADHOC-NOW, LNCS v. 6288*, pages 29–42, 2010.

[7] S. Dobrev, S. Durocher, M. Eftekhari, K. Georgiou, E. Kranakis, D. Krizanc, L. Narayanan, J. Opatrny, S. Shende, and J. Urrutia. Complexity of barrier coverage with relocatable sensors in the plane. In *CIAC*, pages 170–182. Springer, 2013.

[8] M. Eftekhari, E. Kranakis, D. Krizanc, O. Morales-Ponce, L. Narayanan, J. Opatrny, and S. Shende. Distributed local algorithms for barrier coverage using relocatable sensors. In *Proceeding of ACM PODC Symposium*, pages 383–392, 2013.

[9] M. Eftekhari, L. Narayanan, and J. Opatrny. On multi-round sensor deployment for barrier coverage. In *Proceedings of 10th IEEE MASS*, pages 310–318, 2013.

[10] C. F. Huang and Y. C. Tseng. The coverage problem in a wireless sensor network. In *WSNA '03: Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pages 115–121. ACM, 2003.

[11] R. Kapelko and E. Kranakis. On the displacement for covering a square with randomly placed sensors. In *Ad-hoc, Mobile, and Wireless Networks - 14th International Conference, ADHOC-NOW 2015, Athens, Greece, June 29 - July 1, 2015, Proceedings*, pages 148–162, 2015.

[12] R. Kapelko and E. Kranakis. On the displacement for covering a unit interval with randomly placed sensors, Preprint 2014.

[13] E. Kranakis, D. Krizanc, O. Morales-Ponce, L. Narayanan, J. Opatrny, and S. Shende. Expected sum and maximum of displacement of random sensors for coverage of a domain. In *Proceedings of the 25th ACM symposium on Parallelism in algorithms and architectures*, pages 73–82. ACM, 2013.

[14] S. Kumar, T. H. Lai, and A. Arora. Barrier coverage with wireless sensors. In *Proceedings of MobiCom '05*, pages 284–298. ACM, 2005.

[15] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M.B. Srivastava. Coverage problems in wireless ad-hoc sensor networks. In *Proceedings of INFOCOM, vol, 3*, pages 1380–1387, 2001.

[16] M. Mehrandish, L. Narayanan, and J. Opatrny. Minimizing the number of sensors moved on line barriers. In *Proceedings of IEEE WCNC'11*, pages 1464–1469, 2011.

[17] NIST Digital Library of Mathematical Functions. http://dlmf.nist.gov/8.17.

[18] G. Yan and D. Qiao. Multi-round sensor deployment for guaranteed barrier coverage. In *Proceedings of IEEE INFOCOM'10*, pages 2462–2470, 2010.