On Accuracy of Region-based Localization Algorithms for Wireless Sensor Networks

Shigeng Zhang[†‡], Jiannong Cao[‡]
State Key Laboratory for Novel Software Technology[†]
Nanjing University
Nanjing, P.R. China
Email: zsg@dislab.nju.edu.cn,{chenlj,cdx}@nju.edu.cn

Lijun Chen[†],DaoXu Chen[†]
Internet and Mobile Computing Lab[‡]
Hong Kong Polytechnic University
Hong Kong
Email: csjcao@comp.polyu.edu.hk

***Abstract*—Localization is an essential problem in Wireless Sensor Networks (WSNs). Many localization algorithms have been proposed, but few efforts have been paid on theoretical analysis on the accuracy of these algorithms. Because it is naturally to formalize range-based localization problems as deterministic parameter estimation problems, for range-based localization algorithms Cramér-Rao Lower Bound (CRLB) has been used to lower bound the variance on the estimation of sensor's positions. However, few similar works have been done for range-free localization algorithms.**

In this paper, based on geometry properties, we theoretically analyze bounds on accuracy for Region-Based Localization (RBL) algorithms which can be classified as one type of range-free localization algorithms. We prove that if in a RBL algorithm, the deployment region R with the area size s is partitioned into k regions (they can be with any shape and any area size), the localization accuracy is bounded below by $\sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}$, no matter how the algorithm partitions R. Although the lower bound is not theoretically tight, our simulation results show that the gap between this bound and achievable accuracy is very small. We conjecture a tighter lower bound $1.003\sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}$ when k is large enough. We also observe that in order to achieve high localization accuracy, partitioned regions should have nearly the same size. We give three examples with simulation results to show how the results can be used to set the values of the parameters, like k and the corresponding anchor/event number, in a RBL algorithm in order to achieve desired localization accuracy.

I. Introduction

Localization is an essential problem in Wireless Sensor Networks (WSNs). Sensors' accurate position information is necessary in many WSN applications, e.g., environmental data collection, wild animal monitoring, and target tracking [1]–[3]. Accurate position information of sensor nodes is also needed by some geographical routing protocols such as GPSR [4] and GF [5].

Many localization algorithms have been developed to determine the position of nodes in a wireless sensor network, [6]–[18]. According to the mechanisms used in position estimation, these algorithms can be divided into two categories [7]: *range-based* and *range-free*. Range-based localization algorithms [17], [19], [20] need some special hardware to measure the distances (via Time of Arrival (TOA), Time Difference of Arrival (TDOA), Received Signal Strength Indicator (RSSI)) or angles (via Angle of Arrival (AOA)) among sensor nodes. After obtaining enough measurements on distances or angles, these algorithms compute *unknown node*'s positions in the centralized or distributed manners. On the other hand, range-free localization algorithms [6]–[9], [13], [21], [22] require no special hardware and usually use the shortest path between two nodes as the estimation of their distance. In some range-free algorithms a node first determines the region it resides in then uses a point in that region as its position estimation [7], [14]–[16], [23], [24], we call this type of algorithms *Region-Based Localization* (RBL) algorithms. RBL algorithms are usually range-free because they need not to measure distances or angles among nodes.

The ultimate purpose of a localization algorithm is to obtain accurate estimation of the node positions, so it is very important to know how to achieve high localization accuracy in localization algorithms. This motivates people to study the relationship between localization accuracy of and the parameters used in a localization algorithm. Because range-based localization problems can be naturally formalized as deterministic parameter estimation problems, for range-based localization algorithms the classical Cramér-Rao Lower Bound (CRLB) [25] can be used to study the variance of estimation of sensors' positions. CRLBs with respect to different measurements (distances, angles) have been derived [26]–[33]. For range-free localization algorithms, the only work mentioning theoretical analysis on accuracy is [13], in which the authors proposed a method to estimate the best localization accuracy that can be achieved for connectivity-based localization algorithms. The proposed method was later extended to be suitable for two-hop cases in [11] and [12]. However, to the best of our knowledge, no similar works have been done for RBL algorithms.

In this paper, based on geometry properties, we the-

978-1-4244-5113-5/09/$25.00

oretically analyze bounds on localization accuracy for RBL algorithms in an algorithm-independent manner. For a RBL algorithm, assuming all the sensor nodes are uniformly deployed in a convex region R with area size s and R is partitioned into k small regions (they can be with any shape and any area size), we prove that the localization accuracy is bounded below by $\sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}$, no matter how the algorithm partitions R. The only requirement of a divided sub-regions is that its centroid resides in itself. Although the lower bound is not theoretically tight, our simulation results show that the gap between this bound and achievable accuracy is very small. We conjecture a tighter lower bound $1.003\sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}$ when k is large enough and R is in a regular shape. We also observe that in order to achieve high localization accuracy, partitioned regions should have nearly the same size.

The results of our work can be used to set appropriate values of the parameters in a RBL algorithm in order to achieve desired localization accuracy. We give several examples with simulation results to show how the lower bound can be used in RBL algorithms.

The rest of this paper is organized as follows. In Section II we briefly describe a framework of RBL algorithms and derive the expression of localization accuracy in RBL algorithms. In Section III, a lower bound on localization accuracy is derived. We validate our lower bound analysis in Section IV through simulations. The results show that the gap between the lower bound and achievable accuracy is very small. We use several examples to show how to apply our results in configuring RBL algorithms to achieve desired localization accuracy. Then in Section V we introduce related work, including theoretical analysis of localization accuracy for range-based localization algorithms and for connectivity-based localization algorithms. We also briefly discuss the differences between our work and existing works in this section. Finally, we conclude this paper in Section VI.

II. Region-Based Localization algorithms

A. Assumptions

In this paper we assume all sensor nodes are deployed in a convex region R with area size S in 2-dimension space. We assume all nodes (precisely, their *coordinates*) are uniformly distributed in R, say, for any sensor node s we have:

$$p(x,y) = \begin{cases} \frac{1}{S} & (x,y) \in R \\ 0 & \text{otherwise,} \end{cases}$$

where $p(x,y)$ means the probability of s locates at (x,y).

B. RBL algorithms

Algorithm 1 shows a framework of RBL algorithms. Generally speaking, a RBL algorithm consists of two steps. In the first step, it partition the deployment region R into k small regions (It is not necessary that every small region is convex. Here we only require that the centroid of every small region resides in that region). In the second step, each sensor node uses centroid of the region which it resides in as its position estimation. Many localization algorithms can be categorized as RBL algorithms, e.g., [7], [14]–[16], [23], [24].

Algorithm 1 A Framework of RBL algorithms

Step 1: ***Region Partition***
Partition R into k regions $(R_1, R_2, \ldots, R_k)$
Step 2: ***Position Estimation***
for each sensor node s **do**
  if s resides in R_i **then**
    s uses centroid of R_i as its position estimation
  end if
end for

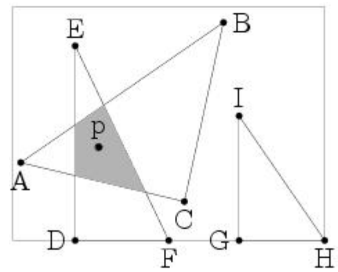

Fig. 1. How PIT works [7]: a) a sensor node s finds that it is in $\triangle ABC$ and $\triangle DEF$, but is out of $\triangle IGH$; b) then s knows it is in the shadowed region and uses p— centroid of that region —as its position estimation.

As an example, we briefly describe the Point-In-Triangle Test (PIT) [7] algorithm here. In PIT, for each triangle formed by three noncollinear anchor nodes, a sensor node tests that if it is in or out of that triangle; then it computes the overlapping region of all triangles where it resides in and uses the centroid of that region as its position estimation. For example, as illustrated in Fig.1, a sensor node s finds that it is in $\triangle ABC$ and $\triangle DEF$ but is out of $\triangle IGH$, then s knows that it is in the region $\triangle ABC \cap \triangle DEF$ and uses the centroid p as its position estimation.

Same as in PIT, many existing RBL algorithms use centroid of the region where a sensor node resides in as its position estimation, but there is no explanation why using the centroid. The reason could be like the following. Intuitively, there are two methods to choose a position estimation for a sensor node. The first is to use a randomly chosen point for each sensor node in a region, and the second is to use same point for all sensor nodes in a region. We argue the first method is not good because: 1) same input should produce same output, so choosing different position estimation for sensor nodes in the same region is not a good idea, and 2) the average location error using the first method may be larger than that of using the second method. For example, the expected distance between two randomly chosen points in a disk with radii r is $\frac{128}{45\pi}r$ [34], while the expected distance between a randomly chosen point in a disk and the centroid of that disk is $\frac{2}{3}r$(which is about .7363 times $128r/45\pi$).

It is not known that if using centroid of a region as position estimation for all sensors in that region will minimize average location error. But using centroid will minimize the variance of estimation of sensors' positions. Let us denote by R_s the region sensor nodes reside in and by S_s its area. Supposing the chosen point is (x_f, y_f), the variance of estimation of sensors' positions is

$$L(x_f, y_f) = \frac{1}{S_s} \iint_{R_s} [(x - x_f)^2 + (y - y_f)^2] \, dxdy,$$

whose minimum value is achieved at

$$\left(\frac{\iint_{R_s} x \, dxdy}{\iint_{R_s} dxdy}, \frac{\iint_{R_s} y \, dxdy}{\iint_{R_s} dxdy}\right).$$

This means that (x_f, y_f) should be R_s's centroid. However, in this paper we still consider using centroid of a region as estimation of position for nodes reside in that region because it is most often used method currently.

It should be pointed out that some literatures use the term "Region-based" to denote localization algorithms which return a region that might contain a sensor node rather than return an exact point [35]–[38]. We do not consider this type of localization algorithms in this paper.

C. Localization accuracy of RBL algorithms

Suppose R with area size s is partitioned into k nonoverlapping regions $R_1, R_2, \ldots, R_k$ with area sizes $s_1, s_2, \ldots, s_k$ by a RBL algorithm. Then the localization accuracy (average location error) is defined as

$$le(R) = \frac{1}{s} \sum_{i=1}^{k} le(R_i) \tag{1}$$

and $le(R_i)$ is defined as

$$le(R_i) = \iint_{R_i} \sqrt{(x - x_c)^2 + (y - y_c)^2} \, dxdy, \tag{2}$$

where (x_c, y_c) is R_i's centroid. Note here that we define $le(R_i)$ as the total location error of all nodes in R_i so there's no coefficient $1/s_i$ in (2).

In this paper, we also consider another metric called *location error variance*, which means variance of estimation of position for all nodes in a region and is defined as

$$lv(R) = \frac{1}{s} \sum_{i=1}^{k} lv(R_i), \tag{3}$$

TABLE I

$e_{sh_{R_i}}$ AND $d_{sh_{R_i}}$ FOR TYPICAL SHAPES

-	Disk	Regular Hexagon	Square	Regular triangle
$e_{sh_{R_i}}$	1	1.003	1.017	1.073
$d_{sh_{R_i}}$	1	1.008	1.047	1.209

where $lv(R_i)$ is defined as

$$lv(R_i) = \iint_{R_i} [(x - x_c)^2 + (y - y_c)^2] \, dxdy. \tag{4}$$

III. Localization accuracy analysis for RBL algorithms

Because the small regions divided by a RBL algorithm can be with any shape and with any area size, it is difficult to calculate (1) and (3) directly. In this section, we first give some preliminary results which can be used to represent $le(R_i)$ and $lv(R_i)$ with only its area size and a factor characterizing its shape, then derive lower bounds on (1) and (3).

A. Preliminaries

In this section we investigate what factors affect the value of $le(R_i)$ and $lv(R_i)$. (For all conclusions here, see the appendix for the proofs.)

1) R_i's position : For two regions R_i and R_j with same shape and same area, it can be proved that $le(R_i) = le(R_j)$ and $lv(R_i) = lv(R_j)$, which means we can translate a region in a 2-dimension space without changing the value of $le(R_i)$ and $lv(R_i)$. For simplicity in presentation, in the rest of the paper we assume all regions locating at the origin, say, the coordinate of R_i's centroid is assumed to be $(0, 0)$.

2) R_i's area: For two regions R_i and R_j with same shape and with area size s_i and s_j respectively, it can be proved that:

$$\frac{le(R_i)}{le(R_j)} = \left(\frac{s_i}{s_j}\right)^{3/2} \text{ and } \frac{lv(R_i)}{lv(R_j)} = \left(\frac{s_i}{s_j}\right)^2. \tag{5}$$

3) R_i's shape: To characterize the affect of R_i's shape on the value of $le(R_i)$ and $lv(R_i)$, we have the following theorem.

***Theorem** 1:* For a region R_i in R^2 which satisfies 1) $\iint_{R_i} dxdy = 1$, and 2) $\iint_{R_i} xdxdy = 0$, $\iint_{R_i} ydxdy = 0$, the minimum value of $le(R_i)$ and $lv(R_i)$ is $2/(3\sqrt{\pi})$ and $\frac{1}{2\pi}$ respectively, and the minimum values are achieved when R_i is a disk.

Proof: See appendix. ■

Then it is obvious that for two regions R_i and R_j with same shape and same area, $le(R_i) = le(R_j)$. Denote by $le(C)$ the value of $le(R_i)$ when R_i is a disk with area

1. For any region R_i with area 1 and with shape sh_{R_i}, define $e_{sh_{R_i}}$ as the ratio between $le(R_i)$ and $le(C)$:

$$e_{sh_{R_i}} = \frac{le(R_i)}{le(C)}. \quad (6)$$

$e_{sh_{R_i}}$ captures the impact of different shapes on $le(R_i)$. Similarly we define $d_{sh_{R_i}}$

$$d_{sh_{R_i}} = \frac{lv(R_i)}{lv(C)}, \quad (7)$$

where $lv(C)$ represents the value of $lv(R_i)$ when R_i is a disk with area 1.

Table I lists values of $e_{sh_{R_i}}$ and $d_{sh_{R_i}}$ for some typical shapes: disk, regular hexagon, square and regular triangle. We can see that $e_{sh_{R_i}}$ and $d_{sh_{R_i}}$ are both small positive numbers slightly greater than 1. The more the region is similar to a disk, the smaller $e_{sh_{R_i}}$ and $d_{sh_{R_i}}$ are. The largest value of these two factors for shapes listed in table I are the values for the regular triangle, which are only 1.073 and 1.209 respectively.

B. A lower bound on localization accuracy for RBL algorithms

Using the results obtained in last subsection, we can rewrite (1) as:

$$le(R) = \frac{1}{s}\sum_{i=1}^{k} s_i^{3/2} e_{sh_{R_i}} le(C) \quad (8)$$

where R_is satisfy 1) $(R_i \cap R_j)_{i \neq j} = \emptyset$, and 2) $\cup_{i=1}^{k} R_i = R$.

Definition 1 (Partition): Given a convex region R, we say a region set $P = \{R_1, R_2, \ldots, R_k\}$ is a *Partition* of R if the regions in P satisfy: 1) centroid of R_i is in R_i; 2) $(R_i \cap R_j)_{i \neq j} = \emptyset$, and 3) $\cup_{i=1}^{k} R_i = R$.

Lemma 1: Given a convex Region R with area s and k positive numbers $s_1, s_2, \ldots, s_k$ whose summation is s, say, $\sum_{i=1}^{k} s_i = s$, there exists at least one Partition of R, $P = \{R_1, R_2, \ldots, R_k\}$, in which area of R_i is s_i.

Proof: See appendix. ■

Given a convex region R, for any partition of R, $\{R_1, R_2, \ldots, R_k\}$, we have

$$le(R) = \frac{1}{s}\sum_{i=1}^{k} s_i^{3/2} e_{sh_{R_i}} le(C) \geq \frac{1}{s}\sum_{i=1}^{k} s_i^{3/2} le(C), \quad (9)$$

where s_i means area of R_i. Using Lemma 1, we know that this inequation holds for all $s_1, s_2, \ldots, s_k$ which subjects to 1) $s_i > 0$ and 2) $\sum_{i=1}^{k} s_i = s$. It is obvious that $le(R)$ is bounded below by (using Lagrange Multiplier Method)

$$(\frac{s}{k})^{1/2} le(C) = \sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}. \quad (10)$$

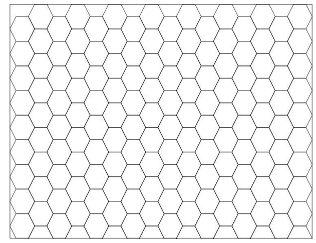

Fig. 2. A conjecture: a tighter lower bound, $1.003\sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}$, may be achieved when k is large enough and R is in a regular shape —use regular hexagons to partition R.

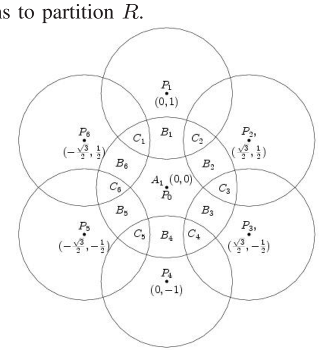

Fig. 3. Test of the closeness of our lower bound to the tight lower bound.

Similarly, the location error variance of all nodes in R is bounded below by

$$\frac{s}{k} lv(C) = \frac{s}{k}\frac{1}{2\pi}.$$

C. A conjecture for a tighter lower bound

The lower bound derived in last section is not theoretically tight, because R cannot be covered by k disks with area s/k. Here we conjecture a tighter lower bound of $le(R)$ when k is large enough and R is in a regular shape.

It is well known that if we want to cover a whole plane using same polygons, only 3 regular polygons can accomplish this task: regular triangle, square, regular hexagon. From Table I we know that regular hexagons have the smallest sh_{R_i} and $d_{sh_{R_i}}$. So we conjecture a tighter lower bound, $1.003\sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}$, may be achieved when k is large enough and R is in a regular shape. This lower bound may be achieved when using regular hexagons to partition R, as shown in Fig.2 .

IV. Validation and Application of the Lower Bound

A. Validation of the lower bound

We conduct some experiments to test how much our lower bound is close to the tight lower bound. We use the algorithm proposed in [15] to execute the test. See Fig.3 for the illustration of the experiment setup.

In Fig.3, $P_0, P_1, \ldots, P_6$ are 7 anchor nodes. $P_1, P_2, \ldots, P_6$ are deployed in a hexagon structure and

Fig. 4. Real localization accuracy vs. the lower bound.

P_0 is deployed at the center of the hexagon. The distance between two adjacent anchor nodes is assumed to be 1. Every anchor node knows all other anchor nodes' positions.

The procedure of the algorithm can be briefly described as following: 1) each anchor node computes the partitioned regions in its transmission range and broadcasts the region list, and 2) each sensor node identifies which region it resides in based on the region lists it received and uses corresponding centroid as its position estimation. For the details please refer to [15].

To fix the deployment region, we set the transmission range of P_0 to be .75 (why we choose this number will be explained later). In order to partition the disk covered by P_0 into 13 regions as shown in Fig.3, we vary the transmission range of $P_1, P_2, \ldots, P_6$ between $[0.52, 0.86]$. We compute average location error of all nodes in the disk covered by P_0 and compare it with our lower bound; the results are shown in Fig.4.

In Fig.4 we can see that the gap between the minimum real average location error and our lower bound is very small, which means that our lower bound is very close to the tight bound (the tight lower bound must be smaller than the minimum real average location error). We also plot the conjectured lower bound, which is nearly the same as our lower bound and is still always smaller than the real average location error.

Our lower bound is obtained when all partitioned regions have same size, so we guess that the more even the partitioned regions are, the smaller the average location error is. To validate this guess, we also plot the *area variance* in Fig.4. In Fig.3 we can see there are 3 types of regions and we identify them with A, B and C respectively. Let $s = \pi * .75^2/13$, the *area variance* is defined as $(s_A - s)^2 + 6*(s_B - s)^2 + 6*(s_C - s)^2$ where s_A, s_B, s_C represents area of A, B, C type of regions respectively. From Fig.4 we can observe that the smaller the area variance is, the smaller the average location error is. This means that the more even the partitioned regions are, the smaller the average location error is,

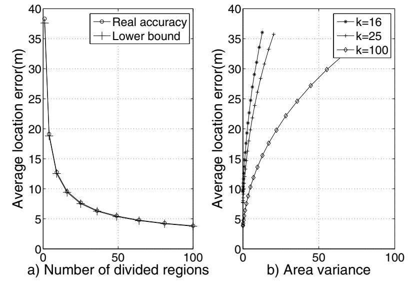

Fig. 5. Factors affecting localization accuracy in a RBL algorithm: number of regions a) and region equality b).

which validates our guess.

B. Factors importance

There are two factors affecting real localization accuracy in a RBL algorithm, k —number of partitioned regions— and the equality of partitioned regions. We conduct two experiments to see how the two factors affect real localization accuracy in a RBL algorithm. In our experiments, 50,000 nodes are uniformly deployed in a 100m*100m region. The region is partitioned into some grids; every node uses the centroid of the grid it residing in as its position estimation. The localization accuracy is calculated as the average of location errors of the 50,000 nodes.

In the first experiment, the deployment region is partitioned equally into $n^2(1 \leq n \leq 10)$ grids. The localization accuracy is shown in Fig.5 a). As we can conclude using formula (10), the larger k is, the lower average location error is. And we can observe that the real localization accuracy is very close the the lower bound, which is proportional to $1/\sqrt{n^2}$. In the second experiment, we fix the number of partitioned regions and vary the area variance. We conduct the second experiment with k set to be 16, 25 ,100 respectively. To make the area variance comparable with each other when k is different, we normalize the area variance by defining it as $\frac{1}{k}\sum_{i=1}^{k}(\frac{s_i}{s'} - 1)^2$ where $s' = s/k$. The results are shown in Fig.5 b). From this fig we can observe that region equality plays more important role than number of regions (k) in achieving high localization accuracy. Even when $k = 100$, with larger area variance the localization accuracy may be lower than that of $k = 16$ with smaller area variance.

C. Application of the lower bound

We use 3 examples to show how to apply our conclusions in RBL algorithms.

1) Optimal transmission range determination: In the algorithm proposed in [15], one problem is to set the optimal transmission range of anchor nodes in order to achieve high localization accuracy. In [15] the optimal transmission range is obtained via simulation. 10, 000

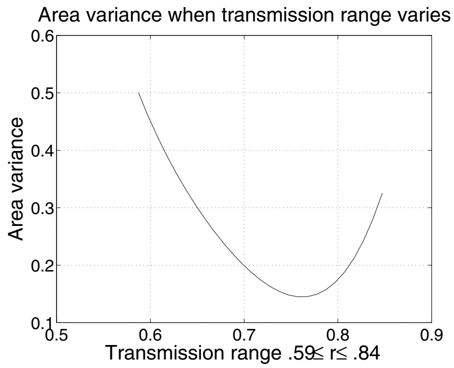

Fig. 6. How area variance changes when transmission range varies.

nodes are used in the simulation and corresponding localization accuracy is computed when the transmission range varies. The authors in [15] found that when the transmission range was 0.744 the localization accuracy was highest.

According to our conclusion, we know that in order to achieve the highest localization accuracy all partitioned regions should have nearly same size. We use area variance as the metric to measure the equality of the size of partitioned regions. Because the area an anchor node covers changes when the transmission range varies and the partitioned regions' shapes also affect localization accuracy, we cannot exactly compute the optimal transmission range. However, we can dramatically reduce the searching space of the optimal transmission range.

Fig.6 shows how area variance changes when the transmission range varies. We can see that the size of partitioned regions are most even when $r \approx 0.757$. Because the area an anchor node covering increases when the transmission range increases, we can conclude that the highest localization accuracy will be achieved when r is less than 0.757. This result is consistent with the result obtained in [15]. However, although the highest localization accuracy is not achieved at $r = 0.757$, at this point it is most close to our lower bound —this is why we set the transmission range of A_0 to be 0.75 in section IV-A .

2) Anchor/event number determination: Recently some "event-driven" localization algorithms have been proposed, [23], [24] and they can actually be categorized as RBL algorithms. In these algorithms, our results can be used to determine how to generate "events" or how to place anchor nodes in order to achieve optimal localization accuracy.

For example, in the basic Multi-Sequence Positioning (MSP) algorithm, only rectangles are used to partition the deployment region, as shown in Fig.7. Suppose the area size of the deployment region is s and the anchor node number is k. The deployment region can be partitioned into at most $(k+1)^2$ regions. As all the partitioned regions are rectangles, we can know that using k anchor nodes the best localization accuracy that can be achieved is $1.017\sqrt{\frac{s}{(k+1)^2}\frac{2}{3\sqrt{\pi}}}$ (in all rectangles, square has the smallest $e_{sh_{R_i}}$). In order to achieve a desired localization accuracy l, k must satisfy

$$l > 1.017\sqrt{\frac{s}{(k+1)^2}\frac{2}{3\sqrt{\pi}}},$$

and we can know that

$$k > \frac{2 \times 1.017 \times \sqrt{s}}{3\sqrt{\pi}l} - 1.$$

If $s = 1000m^2$ and we desire localization accuracy less than $1m$, then $k > 11.1$. This means at least 12 anchor nodes are needed. We can also know that in order to achieve high localization accuracy, the anchor nodes should be placed in a manner in which the deployment region can be evenly partitioned.

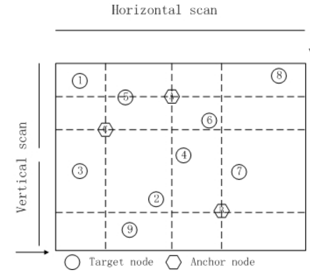

Fig. 7. Basic MSP illustration.

TABLE II
PARAMETERS FOR SIMULATING PIT

Parameter	Value
Deployment region	500m*500m
Node number	2500
Anchor number	[8,26]
Grid granularity	10m
Node's transmission range	50m
Anchor's transmission range	450m
Node deploy strategy	Random

3) RBL algorithms' performance evaluation: The lower bound can be used to evaluate the performance of RBL algorithms, say, how much the real performance of a RBL algorithm is close to the lower bound. As one of the most famous RBL algorithms, PIT [7] is chosen as an example to show how to use the lower bound to do the performance evaluation.

The simulation parameters are listed in Table II. 2,500 nodes are randomly deployed in a 500m*500m region and the number of anchor nodes are tuned between 8 and 26. The transmission range of unknown nodes is set to be 50m and the transmission range of anchor nodes is set to be 450m, resulting in the of *Anchor to Node*

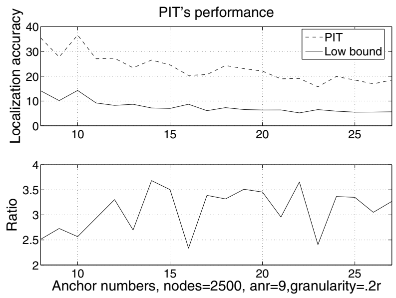

Fig. 8. Performance of PIT [7]: 1) localization accuracy of PIT and lower bound under the same condition, 2) the ratio between them.

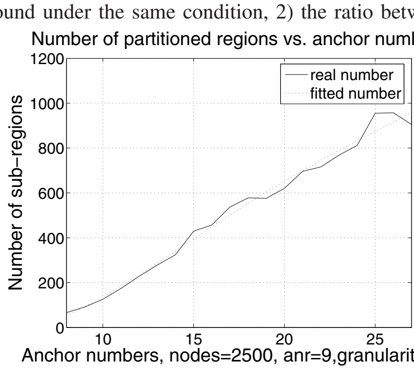

Fig. 9. Relationship between anchor number and partitioned region number. The dashed line shows the fitted value and the real line shows the real value.

Range Ratio (anr) to be 9. The granularity used in the grid scan algorithm [7] is set to be 10m.

We call those nodes which are in at least one triangle *determined nodes*. In the simulation, we only use determined nodes to calculate the localization accuracy of PIT. We count the number of partitioned regions k as the number of distinct position estimations produced by the algorithm (also only count position estimation of determined nodes). To calculate the lower bound, we compute the summation of area of grids there are in at least one triangles formed by 3 noncollinear anchors as the "real" deployment area s. After calculating the lower bound, we compute the ratio p between the real localization accuracy and the lower bound.

The results are shown in Fig.8. In Fig.8 we can find that under the same condition, the localization accuracy of PIT is about 1.5-2.5 times larger than the lower bound.

We also do some simulations to figure out the relationship between the number of partitioned regions and the number of anchors in PIT. In [7] the authors proved that if a sensor node can hear n anchors, the maximum number of polygons partitioned by these anchors is $(n-1)(n-2)/2+n(n-1)(n-2)(n-3)/24$, so we fit $f(n)$ as a function of n whose highest order set to be 4. The result is about $.0044x^4-.35x^3+9.81x^2-62x+84$, which is shown in Fig.9. Using this function and p computed previously, we can know that if we want to achieve localization accuracy less than 25m in a 500m*500m region using PIT, at least 15 anchors are needed. In the simulation the location accuracy is larger than 25m when using only 15 anchors, which verifies this result.

V. Related work

A. Analysis of localization accuracy for Range-based localization algorithms

Range-based localization can be formalized as a classical parameter estimating problem: use $\mathbf{X}$ to estimate θ, where $\mathbf{X}$ is the vector of measurements that are collected and θ is the vector of all the nodes' positions.

We assume a sensor network consists of n sensor nodes $\{s_1, s_2, \ldots, s_n\}$ and denote the position of s_i by (x_i, y_i). Stacking all the nodes' positions we get a vector $\theta = [x_1, y_1, \ldots, x_n, y_n]^T$. Measurements (distances, angles, etc.) used to compute sensor nodes' positions can be defined as a function of θ. For example, the distance d_{ij} between two sensor nodes s_i and s_j is a function of θ:

$$d_{ij} = X(\theta) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}.$$

Suppose m measurements $\{X_1, X_2, \ldots, X_m\}$ are collected. Stacking these measurements we get the vector $\mathbf{X} = [X_1, X_2, \ldots, X_m]^T$.

In real environment, observed measurements usually have errors. We use X to denote the real observed measurements and $p(X;\theta)$ to denote the *PDF (Probability Density Function)* of observing X given θ. If X satisfies the following "regularity" conditions [25]:

$$E[\frac{\partial \ln p(X;\theta)}{\partial \theta_i}] = 0 \text{ for all } \theta_i,$$

then for any unbiased estimator $\hat{\theta}$ of θ, the estimation covariance defined as

$$C(\hat{\theta}) = E\{(\hat{\theta}-\theta)(\hat{\theta}-\theta)^T\}$$

is bounded below by the inverse of the Fisher Information Matrix $I(\theta)$:

$$C(\hat{\theta}) \geq I(\theta)^{-1},$$

where $I(\theta)$ is a $2n \times 2n$ matrix whose elements are defined as

$$[I(\theta)]_{ij} = -E\left[\frac{\partial^2 \ln p(X;\theta)}{\partial \theta_i \partial \theta_j}\right].$$

$I(\theta)^{-1}$ is the CRLB of $C(\hat{\theta})$. After we get $I(\theta)^{-1}$, a lower bound in terms of localization accuracy (average location error) can be computed as

$$\frac{1}{n}\sum_{i=1}^{n}\sqrt{C_{2i-1,2i-1} + C_{2i,2i}}.$$

Some works use CRLB to analyze the localization accuracy for range-based localization algorithms [26]–[28], [30]–[32], [39]. In [27] localization accuracy in localization algorithms using TOA (Time of Arrival) and RSS (Received Signal Strength) is analyzed; in [26] localization accuracy in localization algorithms using distances and angles among sensor nodes is analyzed; and in [28] localization accuracy in localization algorithms using RSS and Quantized RSS is analyzed.

B. Analysis of localization accuracy for connectivity-based localization algorithms

For those localization algorithms using only sensor nodes' local connectivity information, [13] proposed a method to analyze the localization accuracy. The idea is to model the maximum distance a sensor node can move without changing its connectivity status as a random variable Z and compute the expectation of Z. Suppose all sensor nodes are uniformly deployed and a sensor node's average neighbor number is n_{local}. For localization algorithms using only *one-hop* connectivity information, the best localization accuracy that can be achieved is:

$$E(Z) = r\frac{\pi}{4n_{local}},$$

where r represents transmission range of a sensor node. With this method, [11] concluded that the best localization accuracy can be achieved using *one-hop* and *two-hop* connectivity information is about $\pi r/12n_{local}$. The authors of [12] indicated that this lower bound was not rigorous when node density was very low (< 5).

There are some differences between this method and our method. This method applies to algorithms in which a node uses only its "connectivity status" to determine its position. However, in RBL algorithms, a sensor node may not use such information or may use more information in determining its position. For example, in APIT, besides the connectivity status, a sensor also uses its neighbors' received signal strength received from the anchors to determine if its position.

VI. Conclusion

In this paper we derive a lower bound in terms of localization accuracy for RBL algorithms. We identify two factors that may affect localization accuracy in a RBL algorithm: number of partitioned regions k and the equality of partitioned regions. Although the lower bound is not theoretically tight, our simulation results show that it is very close to the tight lower bound. We also observe that the equality of partitioned regions plays more important role than k in achieving high localization accuracy in RBL algorithms. These results can be used to set appropriate values of parameters in RBL algorithms in order to achieve desired localization accuracy and we give three examples to show how to apply our conclusions in RBL algorithms.

Acknowledgement

This work is supported in part by the China Natural Science Foundation under grant 60573132,60873026 and 60573106, the China 973 project under grant 2006CB303000, the Hong Kong RGC under GRF grant PolyU 5/02/08E, and the Hong Kong Polytechnic University under grant 1-BB6C. The authors also gratefully acknowledge the anonymous reviewers' nice comments.

References

[1] J. L. Hill, "System Architecture for Wireless Sensor Networks," Ph.D. dissertation, University of California, Bekerley, 2003.

[2] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring," in *ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02)*, Atlanta, GA, Sep. 2002. [Online]. Available: citeseer.ist.psu.edu/mainwaring02wireless.html

[3] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, "Energy-efficient computing for wildlife tracking: design trdeoffs and early experiences with ZebraNet," in *Proceedings of the 10th international conference on architectural support for programming languages and operationg systems(ASPLOS-X)*, Oct. 2002, pp. 96–107.

[4] B. Karp and H. T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking(Mobicom'00)*. Boston: ACM Press, 2000, pp. 243–254.

[5] J. Navas and T. Imielinski, "GeoCastłgeographic addressing and routing," *Proceedings of the 3rd annual ACM/IEEE international conference on Mobile computing and networking*, pp. 66–76, 1997.

[6] Y. Shang, W. Ruml, Y. Zhang, and F. MPJ, "Localization from mere connectivity," in *Proceedings of the 4th ACM Internatioal Symposium on Mobile Ad Hoc Networking and Computing(Mobihoc'03)*, 2003.

[7] T. He, C. Huang, B. M. Blum, J. A. Stankovic, and T. Abdelzaher, "Range-free localization schemes for large scale sensor networks," in *Proceedings of the 9th Annual International Conference on Mobie Computing and Networking(Mobicom'03)*, 2003, pp. 81–95.

[8] N. Bulusu, J. Heidemann, and D. Estrin, "GPS-less low cost outdoor localization for very small devices," *IEEE Personal Communications*, vol. 7, no. 5, pp. 28–34, Oct. 2000.

[9] D. Niculescu and B. Nath, "Ad-hoc positioning system(aps)," in *Proceedings of 2001 IEEE Global Telecommunications Conference(Globecom'01)*. San Antonio, TX, USA: IEEE Communications Society, May 2001, pp. 2926–2931.

[10] M. Li and Y. Liu, "Rendered path: Range-free localization in anisotropic sensor networks with holes," in *Proceedings of the 13th annual ACM international conference on Mobile computing and networking(Mobicom'07)*. Montreal, Queébec, Canada: ACM, Sep. 2007, pp. 51–62.

[11] L. Hu and D. Evans, "Localization for mobile sensor networks," in *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking(Mobicom'04)*. Philadelphia, Pennsylvania: ACM Press, 2004, pp. 45–57.

[12] M. Rudafshani and S. Datta, "Localization in wireless sensor networks," in *Proceedings of the 6th International Conference on Information Processing in Sensor Networks(IPSN'07)*. Cambridge, Massachusetts, USA: ACM, 2007, pp. 51–60.

[13] R. Nagpal, H. Shrobe, and J. Bachrach, "Organizing a global coordinate system from local information on an ad hoc sensor networks," in *Proceedings of the 2nd International Workshop on Infromation Processing in Sensor Networks(IPSN'03)*, F. Zhao and L. J. Guibas, Eds. Springer-Verlag, 2003, pp. 333–348.

[14] L. Lazos and R. Poovendran, "Serloc: Secure range independent localization for wireless sensor networks," in *Proceedings of 3rd ACM workshop on wireless security*. Philadelphia, PA, USA: ACM, 2004, pp. 21–30.

[15] H.-C. Chu and R.-H. Jan, "A gps-less, outdoor, self-positioning method for wireless sensor networks," *Ad Hoc Networks*, vol. 5, no. 5, pp. 547–557, 2007.

[16] C. Liu and K. Wu, "Performance evaluation of range-free localization methods for wireless sensor networks," in *Proceedings of the 24th International Conference on Performance, Computing, and Communications(IPCCC2005)*, Apr. 2005, pp. 59–66.

[17] D. K. Goldenberg, P. Bihler, M. Cao, J. Fang, B. D. Anderson, A. S. Morse, and Y. R. Yang, "Localization in sparse networks using sweeps," in *Proceedings of the 12th annual international conference on mobile computing and networksing (MobiCom'06)*. ACM, 2006, pp. 110–121.

[18] Z. Shigeng, J. Cao, C. Lijun, and C. Daoxu, "Location nodes in mobile sensor networks more accurately and faster," in *secon08*, 2008.

[19] A. Savvides, C.-C. Han, and M. B. Srivastava, "Dynamic fine-grained localization in ad-hoc sensor networks," in *Proceedings of the 7th Annual international Conference on Mobile Computing and Networking(Mobicom'01)*. Rome, Italy: ACM, 2001, pp. 166–179.

[20] J. Liu, Y. Zhang, and F. Zhao, "Robust distributed node localization with error management," in *Mobihoc'06*, May 2006.

[21] Y. Shang and W. Ruml, "Improved MDS-Based localization," in *Proc. of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies(Infocom'04)*, 2004, pp. 2640–2651.

[22] X. Ji and H. Zha, "Sensor positioning in wireless ad-hoc sensor networks using multidimensional scaling," in *Proc. of 23rd Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM'04)*, 2004, pp. 2652–2661.

[23] Z. Zhong and T. He, "Msp: multi-sequence positioning of wireless sensor nodes," in *Proceedings of the 5th international conference on embedded networked sensor systems (Sensys'07)*, 2007, pp. 15–28.

[24] Z. Zhong, D. Wang, and T. He, "Sensor node localization using uncontrolled events," in *International Conference on Distributed Computing Systems (ICDCS 2008)*, Jun. 2008.

[25] S. M. Key, *Fundamentals of Statistical signal processing: Estimation Theory*. PTR Prentice Hall, 1993, vol. I.

[26] A. Savvides, W. L. Garber, R. L. Moses, and M. B. Srivastava, "An analysis of error inducing parameters in multihop sensor node localization," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. 4, no. 6, pp. 567–577, 2005.

[27] N. Patwari, A. O. H. III, M. Perkins, N. S. Correal, and R. J.O'Dea, "Relative location estimation in wireless sensor networks," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 51, no. 8, pp. 2137–2148, Aug. 2003.

[28] N. Patwari and A. Q. H. III, "Using proximity and quantized rss for sensor localizations in wireless networks," in *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, 2003, pp. 20–29.

[29] C. Chang and A. Sahai, "Estimation bounds for localization," in *Proceedings of 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks(SECON'04)*, Oct. 2004, pp. 415–424.

[30] R. L. Moses, D. Krishnamurthy, and R. M. Patterson, "A self-localization method for wireless sensor networks," *EURASIP Journal on Applied Signal Processing*, no. 4, pp. 348–358, 2003.

[31] D. Ştefan Niculesce, "Forwarding and positioning problems in ad hoc networks," Ph.D. dissertation, State University of New Jersey, 2004.

[32] Y. Shang, H. Shi, and A. A.Ahmed, "Performacne study of localization methods for ad-hoc sensor networks," in *Proc. of 1st IEEE Conference on Mobile Ad-hoc and Sensor Systems(MASS 2004)*, oct 2004, pp. 184–193.

[33] D. Niculescu and B. Nath, "Error characteristics of ad hoc positioning systems (aps)," in *MobiHoc '04: Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing*. New York, NY, USA: ACM, 2004, pp. 20–30.

[34] L. Kleinrock and J. Silverster, "Optimum transmission radii for packet radio networks or why six is a magic number," in *Proc. of the IEEE National Telecommunications*, Dec. 1978, pp. 4.3.1–4.3.5.

[35] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: a comparative study," in *Proceedings of the 1st Annual IEEE Communications Society Coference on Sensor and Ad Hoc Communications and Networks(SECON'04)*, Oct. 2004, pp. 406–414.

[36] Z. Feng, W. Liu, K. Xu, and W. Cheng, "Robust region based localization for practical sensor networks," in *Proceedings of Wireless Communications and Networking Conference (WCNC) 2007*, Mar. 2007, pp. 3670–3674.

[37] E. Elnahrawy, X. Li, and R. P. Martin, "Using area-based presentations and metrics for localization systems in wireless lans," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks(LCN'04)*, 2004.

[38] F. Wang, L. Qiu, and S. S.Lam, "Probabilistic region-based localization for wireless networks," *ACM SIGMOBILE Mobile Computing and Communications Review, SPECIAL ISSUE: Special issue on localization*, vol. 11, pp. 3–14, Jan. 2007.

[39] A. Savvides, W. Garber, S. Adlakha, R. Moses, and M. B.Srivastava, "On the error characteristics of multihop node localization in ad-hoc sensor networks," in *Proceedings of the 2nd International Workshop on Infromation Processing in Sensor Networks(IPSN'03)*, F. Zhao and L. J. Guibas, Eds. Springer-Verlag, 2003, pp. 317–332.

[40] "http://en.wikipedia.org/wiki/centroid."

Appendix

A. Proof for Preliminaries

For R_i***'s position***: Suppose R_i's centroid is (x_i, y_i) and R_j's centroid is (x_j, y_j), let $R_i' = \{(u,v)|u = x - x_i, v = y - y_i, (x,y) \in R_i\}$ and $R_j' = \{(u,v)|u = x - x_j, v = y - y_j, (x,y) \in R_j\}$, then we have

$$\begin{aligned} le(R_i) &= \iint_{R_i} \sqrt{(x-x_i)^2 + (y-y_i)^2}\, dxdy \\ &= \iint_{R_i'} \frac{\partial(x,y)}{\partial(u,v)} \sqrt{u^2+v^2}\, dudv \\ &= \iint_{R_i'} \sqrt{u^2+v^2}\, dudv = le(R_i') \qquad (11) \end{aligned}$$

and it is obvious that $le(R_j) = le(R_j')$ and $R_i' = R_j'$. So we have $le(R_i) = le(R_j)$. For $lv(R_i)$ the proof is similar.

For R_i***'s area***: Let $D1 = \{(x,y)|(x,y) \in R_i\}$ and $D2 = \{(u,v)|(u,v) \in R_j\}$. Using previous result, we can assume the centroid of $D1$ and $D2$ are both at $(0,0)$. Then we have

$$\begin{aligned} le(R_i) &= \iint_{D1} \sqrt{x^2+y^2}\, dxdy \\ &= \iint_{D2} \frac{\partial(x,y)}{\partial(u,v)} \sqrt{\frac{s_i}{s_j}u^2 + \frac{s_i}{s_j}v^2}\, dudv \\ &= (s_i/s_j)^{3/2} le(R_j). \qquad (12) \end{aligned}$$

For $lv(R_i)$ the proof is similar.

For ***Theorem 1***:

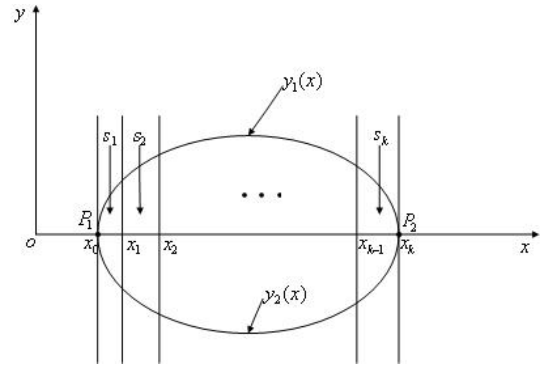

Fig. 10. Given an area condition $(s_1, s_2, \ldots, s_k)$, there exists at least one partition of R satisfies that condition —use parallels to partition R.

Proof: We first prove the minimum value of $\iint_{R_i} \sqrt{x^2+y^2}dxdy$ is achieved when R_i is a disk, then calculate the minimal value.

Let $C = \{(x,y) : x^2+y^2 \leq r^2\}$, where $\pi r^2 = 1$, then C is a disk satisfing conditions 1) and 2) listed in Theorem 1. Let

$$le(C) = \iint_C \sqrt{x^2+y^2}dxdy.$$

For an arbitrary region R_i which satisfies the two conditions, we have:

$$R_i = (R_i \cap C) \cup (R_i \backslash C) \text{ and } C = (C \cap R_i) \cup (C \backslash R_i).$$

So

$$le(R_i) - le(C) = \iint_{R_i \backslash C} \sqrt{x^2+y^2}dxdy - \iint_{C \backslash R_i} \sqrt{x^2+y^2}dxdy \geq r(\iint_{R_i \backslash C} dxdy - \iint_{C \backslash R_i} dxdy).$$

Using condition 1) we know that

$$\iint_{R_i \backslash C} dxdy - \iint_{C \backslash R_i} dxdy = 0,$$

and so we have

$$le(R_i) \geq le(C).$$

Since R_i is arbitrary, we prove that formula $\iint_{R_i} \sqrt{x^2+y^2}dxdy$ achieves its minimum value when R_i is a disk, and the value is $2/(3\sqrt{\pi})$, which can be be easily computed. For $lv(R_i)$ we have similar proof and its minimum value is $\frac{1}{2\pi}$. ■

B. Proof for Lemma 1

As shown in Fig.10, we can use parallels to partition R into k small regions. Let P_1, P_2 be two points in R such that $|P_1P_2| \geq |P_iP_j|, \forall P_i, P_j \in R$. We then move R by translation and rotation to make P_1 and P_2 both in the x-axis, as shown in Fig.10.

Without loss of generality, we use $y_1(x)$ and $y_2(x)$ to describe the two curves enclosing R, as shown in Fig.10. To obtain a partition satisfying the area condition $(s_1, s_2, \ldots, s_k)$, we use $k-1$ vertical lines, $x = x_i, 1 \leq i \leq k-1$, to partition R. $x = x_i, 1 \leq i \leq k-1$ should satisfy the following equations:

$$\int_{x_{i-1}}^{x_i}\int_{y_2(x)}^{y_1(x)} dxdy = s_i, 1 \leq i \leq k. \quad (13)$$

Because we know x_0 and x_k, we can calculate x_i iteratively. For example, for x_1 we have:

$$\int_{x_0}^{x_1} \left[y_1(x) - y_2(x)\right] dx = s_1.$$

Because both $y_1(x)$ and $y_2(x)$ are continuous, there must be a $F(x)$ such that $F' = y_1(x) - y2(x)$, and we have $F(x_1) = s_1 + F(x_0)$. So we can find a solution of x_1, and do the same to get $x_2, x_3, \ldots, x_{k-1}$.

It is obvious that each partitioned region R_i is convex, so for each R_i its centroid is in R_i [40]. So completed the proof.

C. Proof for the lower bound

It is enough to find a lower bound for $\frac{1}{s}\sum_{i=1}^{k} s_i^{3/2} le(C)$ which subjects to $\sum_{i=1}^{k} s_i = s, s_i > 0, 1 \leq i \leq k$.

Construct a new function with same value:

$$F(s_1, s_2, \ldots, s_k) = \sum_{i=1}^{k} s_i^{3/2} + \lambda(\sum_{i=1}^{k} s_i - s). \quad (14)$$

(because $le(C)$ and s are all constants, we omit these two coefficients in the derivation). Using Lagrange Multiplier Method, F's minimum value is achieved when

$$\frac{\partial F}{\partial s_i} = 0 = \frac{3}{2}s_i^{1/2} + \lambda,\ 1 \leq i \leq k. \quad (15)$$

Then we have

$$\lambda = -\frac{3}{2}\sqrt{\frac{s}{k}}$$

and F's minimum value is

$$\sum_{i=1}^{k} s_i^{1/2} s_i = -\frac{2\lambda}{3}\sum_{i=1}^{k} s_i = s\sqrt{\frac{s}{k}}.$$

So we have

$$\frac{1}{s}\sum_{i=1}^{k} s_i^{3/2} le(C) \geq \sqrt{\frac{s}{k}}\frac{2}{3\sqrt{\pi}}.$$

For location error variance, the proof is similar.