Approximate coverage technique of WSN design

Levon Aslanyan
Institute for Informatics and Automation Problems of NAS RA
lasl@sci.am

Hakob Aslanyan
Institute for Informatics and Automation Problems of NAS RA
hakob.aslanyan@gmail.com

ABSTRACT

Paper provides technical means of digitized approximation of sensing area coverage subsystem in WSN (wireless sensor networks). Approximate coverage versus to complete cover is acceptable for some applications and the use of approximations can extend the network lifetime. The digitized approximation is considered as an alternative to the analytic, set cover, Voronoi diagram based and other techniques, that are in use today. Parameters of accompanying structures were studied and estimated.

Keywords

WSN, coverage, digitization, quad tree.

1. INTRODUCTION

We consider the algorithmic problem of covering the plain monitoring areas given in WSN systems. The disk model of WSN is supposed for a fixed radius r. A target coverage area $A$is considered under the general supposition of convexity. The set of sensor nodes over the domain A may compose a reachable infrastructure that is able to cover each point $a \in A$ many times. There is a broad literature that considers the exact coverage framework. Two postulations are typical here: (1) check for a given set of sensor nodes if in their union they cover the target area A, and (2) split, because of the redundancy, the sensor nodes into the groups and check if each of them provides the exact coverage of the target area A. In technical level it is important to know the format/structure, in which the area A is given. In general this can be an analytic expression; otherwise it can be given by an algorithmic data structure. Simplest is the polygonal area A given by the set of corner points. Expressed analytically coverage deals with specific sets of equations – mostly with linear equations. The algorithmic data structure formats appear in the digitized models, when partitions of coverage area into smaller parts are considered like in triangulations, etc.

A special very important technique of target domain coverage is related to the Voronoi diagrams. A larger (but restricted) part of the space containing coverage area A is split into the domains according to the given set of sensor nodes. Partition domains correspond to sensors that are situated in those domains. Such sensors are the first candidates to be considered as covering. But one sensor, with large sensing radius r, can cover not only its own domain but also several neighbouring domains. Then the neighbouring sensors become not necessary for the coverage at this point. The game (node deletion, insertion) with domain restructuring is a local and algorithmically not a complex task. Voronoi diagram itself is a polynomial complexity algorithmic problem.

The novelty considered in this paper is related to the fact that several WSN applications are not critical in the sense of an exact coverage. Approximate coverage works with set differences that appear around the boarder of the base area A. As an alternative to the analytic or to data structure models, the digitized version of the framework is considered, and simple relations between the model parameters were derived. The result is a technical framework that may be productive in WSN design with an approximate coverage subsystem. The real size implementation still depends on particularities of the target area A, and the properties of the system, like the coverage number required (multiplicity), and others.

2. APPROXIMATE COVER BY CIRCLES

Let k denote the required number of covers of elements $a \in A$ by sensors.As the area A is a continuous region (as a rule), it becomes impossible to check this coverage condition

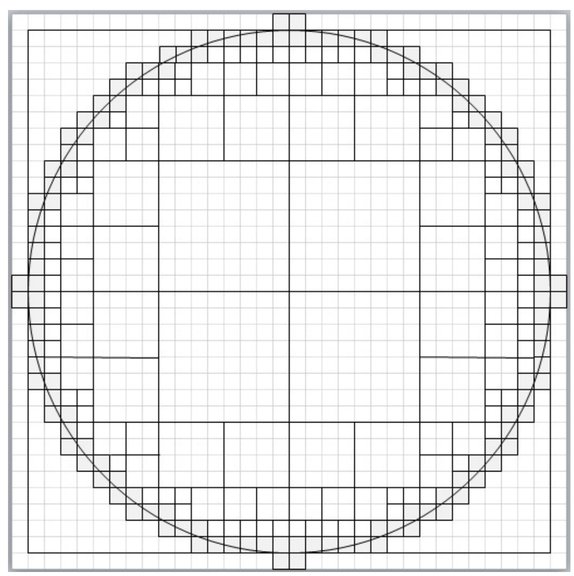

Figure 1. Recursive splitting of rectangles

for all points of A, therefore, we bring the coverage constraint to a discrete domain, by generating a grid structure over the sensing area, and then check the coverage condition for the grid cells (rectangles) only (Figure 1). As a result of such transformation some of the grid cells become partially covered (those cells that intersect with the circles of the sensing discs), and to guarant a perfect coverage of a given region one should consider partially covered cells as uncovered (even if in reality the union of several such discs may cover these cells) and seek for other sensing discs that cover these cells entirely. Below we describe a recursive algorithm for transforming a continuous area into a grid structure and give an upper bound for partially covered cells in terms of their surfaces.

First we describe an algorithm that builds a grid structure over a rectangular area in a presence of a single sensing disc (Figure 1). This structure is later used for error approximations (partially covered cells).

Let a be the sensing disc of a sensor node s and r be the minimal bounding rectangle of a. We build a so-called quad tree in the following way. We split the root rectangle r into four equal rectangles (Figure 1), this creates four leaves that are childe nodes of the root rectangle r. We apply the splitting procedure recursively to each newly created leaf rectangle until the new rectangles reach certain size. We are interested in leaf rectangles that have only one, two or three corner points inside the disc a. These rectangles approximate the border of the disc and considered as uncovered while in reality they are partially covered. Leaf rectangles that have four corner points inside the disc are completely covered by the disc and recursive algorithm will not be applied to those leaves. Rectangles with zero corner points inside the disc form the outer part of the disc and the recursive algorithm will not be applied to those rectangles, too.

To check if a corner point *p* of a given rectangle belongs to the disc *a* we simply calculate the distance d(p, a) of p from the center of a and check if it is not greater than the radius R_s of a. i.e. we check the following

$$(x_p - x_a)^2 + (y_p - y_a)^2 \leq R_s^2$$

where (x_p, y_p) are the coordinates of p and (x_a, y_a) are the coordinates of the center of a.

We define a function $t_1: \mathbb{R} \times \mathbb{D} \to \{0,1,2,3,4\}$ where $\mathbb{R}$ is the space of rectangles, $\mathbb{D}$ is the space of discs and $t_1(r, a)$ is the number of corner points of r that are covered by a.

Quad Tree Algorithm (QTA)

//TODO: PSEUDOCODE

We use a version of the Quad Tree algorithm to check the coverage constraint. We say a rectangle is covered if there is a single sensing disc that covers it entirely.

Assume a set of l sensor nodes are deployed in a two-dimensional plane and we want to check if a rectangular region A with sidesL_x and L_y is covered by the sensing discs of the given sensor nodes. Without loss of generality we will assume that $L_x = L_y = L$ (otherwise we could divide A into square-like regions and check the coverage condition for each region separately). To answer the coverage question we build a grid structure over A by applying a quad tree-building algorithm similar to the one discussed above. The algorithm starts with a root rectangle A and splits it into four equal rectangles generating four leaf rectangles. Further the algorithm is applied recursively to all the leaf rectangles that intersect with at least one sensing circle (boundary of sensing disc) and are not contained in any of sensing discs (not covered). We call this a *splitting condition*. For checking the *splitting condition* for a given rectangle r we first determine the number $t_1(r, a_i)$ for all the sensing discs $a_i, i = 1, \dots, l$ and do the following checks:

C1. If $\exists j \in \{1, \dots, l\}$ such that $t(r, a_j) = 4$ then the rectangle r is covered by a_j and does not satisfy the *splitting condition.*

C2. Given that the first condition is not satisfied we check if $\exists j \in \{1, \dots, l\}$ such that $1 \leq t(r, a_j) \leq 3$ then the rectangle r intersects/approximates the circle of the sensing disc a_j and satisfies to *splitting condition.*

The algorithm finishes its work in the following cases:

E1. ∃a leaf rectangle r such that $t(r, a_j) = 0$ for $\forall j \in \{1, \dots, l\}$, i.e. all the corner points of r are out of coverage. In this case the algorithm finishes its work and returns *uncovered*.

E2. There is no leaf rectangle that satisfies the *splitting condition*, i.e. all the rectangles are covered. In this case algorithm finishes its work and returns *covered.*

E3. The height of the quad-tree becames some predefined number k (equivalently, all the leaf rectangles satisfying the *splitting condition* reached a certain size), i.e. no uncovered rectangle has been found so far (all the rectangles are either covered or partially covered). In this case the algorithm finishes its work and returns *almost covered* along with an upper bound of partially covered rectangles in terms of their surfaces (discussed below).

Extended Quad Tree Algorithm (EQTA)

A rectangle can be given by a set of its corner points or by one corner/center point and the level it has in the quad tree, other three coordinates can be derived from these as all the rectangles on the m-th level of the quad-tree have sides $\Delta = L/2^m$.

//TODO: PSEUDOCODE EXTENDED

The above algorithm can be easily adopted for a K-coverage case by replacing the conditions $C1$ and $E1$ with $C1'$ and $E1'$, respectively.

C1'. If $\exists j_1, \dots, j_K \in \{1, \dots, l\}$ such that $t(r, a_{j_i}) = 4$ for all $i \in \{1, \dots, K\}$ then the rectangle r is covered by the discs $a_{j_1,\dots,}a_{j_K}$ and does not satisfy the *splitting condition.*

D1'. ∃a leaf rectangle r and discs$a_{j_1,\dots,}a_{j_{l-K+1}}$ such that $t(r, a_{j_i}) = 0$ for all $i \in \{1, \dots, l - K + 1\}$, i.e. there are $l - K + 1$ circles that obviously do not cover endpoints of r, and even if each of the remaining $K - 1$ discs covers rK-coverage cannot be guaranteed.

3. TERMS OF MINKOWSKI GEOMETRY

The well-known Minkowski technique for approximation of convex bodies [1,§17] complements our algorithmic constructions and gives a guaranteed approximation accuracy for *partially covered* (EQTA-E3) rectangular cells in terms of their surface. Below we introduce the basic notations of Minkowski geometry and give the upper bound for the *partially covered* cells in the next section.

For an arbitrary convex figure F consider another figure G that extends F. Particular extensions are of interest:

- **λ-multiple**$G = \lambda F$ is a figure composed after multiplication of coordinates of vertices of F by a coefficient λ, $\lambda F = \{\lambda x \mid x \in F\}$. This transformation does not depend on the coordinate system itself and provides a λ enlargement of F. For surfaces $O_{\lambda F}$ and O_F of λF and F the following holds $O_{\lambda F} = \lambda^2 O_F$. This relation is easy to prove for a polytope but it is extendable to the general case of convex bodies as well.

- **Another important transformation** of F is the figure $G = F_\rho$, which consists of all the points that are not farther than ρ from F. F_ρ composes a convex figure that is called to be ***parallel to F***. F_ρ can change the shape of F. Surfaces of F and F_ρ obey the general relation $O_F \leq O_{F_\rho}$. A more specific relation between the pair O_F and O_{F_ρ} will be reduced below.

Definition 1: For figures A and B the set $A ‡ B = \{x + y \mid x \in A, y \in B\}$ is called **Minkowski sum** of A and B. Here are some initial properties with Minkowski sum ‡

- $A \subseteq B \Rightarrow A ‡ C \subseteq B ‡ C$
- If A, B are convex, then $A ‡ B$ is convex
- $(A \cup B) ‡ C = (A ‡ C) \cup (B ‡ C)$

More properties relay Minkowski sum with λ-multiple

- $\alpha(A ‡ B) = \alpha A ‡ \alpha B$,
- $(\alpha + \beta)A \subseteq \alpha A ‡ \beta A$, and if A is convex, then equality holds.

Lemma 1: Let s_ρ be a sphere of diameter ρ and F is a convex figure, then $F_\rho = F ‡ s_\rho$.

Proof: From Definition 1 it follows that $F ‡ s_\rho$ contains points that have a distance at most ρ from F. On the other hand for any point z that has a distance ρ or less from F there is a point $x \in F$ such that $d(z, x) \leq \rho$ and therefore $z \in s_\rho ‡ \{x\} \subseteq F ‡ s_\rho$.

Consider a line L on the plane, representing it as $L = \{x \in R^2 : a^t \cdot x = \alpha\}$ for some nonzero vector a and a real number α. $\alpha = 0$ determines the line perpendicular to the vector a. Then the vector a is called the normal vector of the line. Each line divides the plain into two sets $L_+ = \{x \in R^2 : a^t \cdot x \geq \alpha\}$ and $L_- = \{x \in R^n : a^t \cdot x \leq \alpha\}$. Half-planes L_+ and L_- intersect in L. If convex figure F is contained in one of the half-planes L_+ or L_- and $L \cap F$ is nonempty, we say that L is a supporting line to F. Each boundary point of F belongs at last to one support line and the figure itself lies on one side of that line.

Set up the unit vector a outgoing from (x_i, y_i) (center of coordinates, and an internal vertex of F), that is perpendicular to the support line L to F (at some point x), and let the direction cosines of a be β and γ. Determine the ***support function*** by Minkowski to the closed convex figure F

$$h(a) = \sup \{a^t \cdot x, x \in F\}.$$

$h(a) \equiv h(\beta, \gamma)$ is the distance of support line L perpendicular to a, to the center of coordinates (x_i, y_i). Similarly we consider the boarder function $r(a)$ that is the distance of intersection of direction a and boundary of F, to the center of coordinates (x_i, y_i). To better understand $h(a)$ and $r(a)$ it is to consider and analyze them on a disc figure or/ and on a triangle (both convex). If we apply addition or multiplication on $r(a)$, then the figure modified can change its shape. So λF, for example, can't be achieved in this way.

The support function allows us to compute the Minkowski sum of convex sets in a very intuitive way. The Minkowski sum can be represented by the sum of support functions; λ-multiple will be represented by λ-multiple of a support function. Moreover, we have an analytical description of subset relation. The subset relation will be represented by comparison of support functions. It is not even needed to handle all x elements, but only $x \in s_\rho$.

1. $h_{\lambda A} = \lambda h_A$,
2. $h_{A‡B} = h_A + h_B$,
3. if$A \subseteq B$, then $h_A \leq h_B$,
4. for$A$ and B closed, $A \subseteq B$ if and only if $h_A \leq h_B$, and $A = B$, if and only if $h_A = h_B$.

Coverage approximation

Here we discuss the exit E3 of Extended Quad Tree Algorithm and give an upper bound for a coverage error in terms of surfaces. Let a_s be the sensing disc of a node s and have a diameter r_s, and let the algorithm be interrupted after k recursive rounds, i.e. the smallest rectangles in the resulting quad tree have sides $\Delta = L/2^k$. Denote by R_s^I the body composed of all the recursive rectangles that are contained in a_s. The coverage error can be estimated in different ways. One of the possible schemes is as follows. Theoretically the sensing surface of node s is $O(a_s) = \pi r_s^2$ while the area covered by s given by Quad Tree Algorithm is R_s^I, therefore, the coverage error is $O(a_s \backslash R_s^I)$ and, of course, it hardly depends on Δ and the coordinates and the radius of s. We do not hope to describe the error in exact terms but we will give an upper bound. Consider a disc $a_{s'}$ centered at the center of s and having a radius $r_{s'} = r_s - \rho$, where $p = \Delta\sqrt{2} = \sqrt{2} \cdot L / 2^k$ is the diameter of the smallest recursive rectangle (note that all the rectangles intersecting with the circle of a_s are of smallest size $\Delta \times \Delta$). It is obvious that $a_{s'} \subset R_s$ and therefore $a_s \backslash R_s \subset a_s \backslash a_{s'}$ and I

$$e_k = O(a_s \backslash a_{s'}) = \pi r_s^2 - \pi r_{s'}^2 = \pi(r_s^2 - r_s^2 + 2r_s\rho - \rho^2) < 2\pi r_s \rho = \frac{\sqrt{2} \cdot \pi r_s L}{2^{k-1}} \quad (*)$$

is an upper bound for the coverage error of the disc a_s.

The following theorem can be formulated.

Theorem 1

For any $\varepsilon > 0$ the coverage error of the Quad Tree Algorithm for a single sensing disc given in an $L \times L$ area can be bounded by ε with a $k = \left\lceil \log\left(\frac{\sqrt{2}\cdot\pi r_s L}{\varepsilon}\right)\right\rceil + 1$ depth quad tree.

Proof

The proof directly follows from (*).

Corollary 1.1

The coverage error of l sensing discs given in an$L \times L$ area can be bounded by ε with a $k = \left\lceil \log\left(\frac{\sqrt{2}\cdot\pi L}{\varepsilon}\left(\sum_{i=1}^{l} r_i\right)\right)\right\rceil + 1$ depth quad tree. The proof follows from (*) and the fact that if $O(a_i \backslash a_{i'})$ is a coverage error of a disc a_i then the surface of $\cup_{i=1}^{l}(a_i \backslash a_{i'})$ will be bounded by $\sum_{i=1}^{l} O(a_i \backslash a_{i'})$. In case of uniform sensing radiuses the formula will be $k = \left\lceil \log\left(\frac{\sqrt{2}\cdot\pi l r L}{\varepsilon}\right)\right\rceil + 1 = \left\lceil \log(l) + \log\left(\frac{\sqrt{2}\cdot\pi r L}{\varepsilon}\right)\right\rceil + 1$.

Corollary 1.2

For any $\varepsilon > 0$ the coverage error of the Quad Tree Algorithm for a single sensing disc can be bounded by ε with a $k = \left\lceil \log\left(\frac{4\sqrt{2}\cdot\pi r_s^3}{\varepsilon}\right)\right\rceil + 1$ depth quad tree. This is the case when the initial bounding rectangle is the minimal rectangle ($L = 2r_s$) that contains the disc a_s.

Theorem 2

The number of **recursive calls** to the Quad Tree algorithm for bounding the coverage error of a single sensing disc a_s by $\varepsilon > 0$, given in an $L \times L$ area, is bounded by $\frac{1}{3}\left(\frac{4\pi r_s L}{\varepsilon}\right)^2$.

Proof

From **Theorem 1** we have that a quad tree of height $k = \left\lceil \log\left(\frac{\sqrt{2}\cdot\pi r_s L}{\varepsilon}\right)\right\rceil + 1$ can bound the coverage error of a single sensing disc (given in an $L \times L$ area) by ε. In the worst case a perfect quad tree of height k should be constructed. These will lead to a tree of $\sum_{i=0}^{k} 4^i$ nodes and for constructing the m-th level of a tree the QT algorithm should be applied to each node situated at the $(m-1)$-th level of a tree, i.e. the number of QT - calls is bounded by $H(\varepsilon, a_s) = \sum_{i=0}^{k-1} 4^i = (2^{2k-1} - 2)/3$. By replacing k with $\log\left(\frac{\sqrt{2}\cdot\pi r_s L}{\varepsilon}\right) + 2$ and simplifying the resulting formula we get

$$H(\varepsilon, a_s, L) < \frac{1}{3}\left(\frac{4\pi r_s L}{\varepsilon}\right)^2$$

Corollary 2.1

For l sensing discs the number of recursive calls is bounded by

$$\frac{1}{3}\left(\frac{4\pi L}{\varepsilon}\sum_{i=1}^{l} r_i\right)^2$$

follows from **Corollary 1.1.** In case of uniform sensing radiuses the bound will be

$$\frac{1}{3}\left(\frac{4\pi l r L}{\varepsilon}\right)^2$$

Corollary 2.2

For a single sensing disc a_s given in the minimal bounding rectangle ($L = 2r_s$) the following is true

$$H(\varepsilon, a_s, 2r_s) < \frac{1}{3}\left(\frac{8\pi r_s^2}{\varepsilon}\right)^2 = \frac{64}{3\varepsilon^2}\cdot(\pi r_s^2)^2 = \frac{64}{3\varepsilon^2}\cdot(O(a_s))^2$$

where $O(a_s)$ is the surface of the disc a_s.

From **Corollaries 1.1 and 2.1** it follows that the computational complexity of the Quad Tree algorithm for bounding a coverage error of l sensing discs by $\varepsilon > 0$ is the same as for bounding the error of a single disc with radius $\sum_{i=1}^{l} r_i$.

4. ACKNOWLEDGEMENT

The authors would like to thank FP7 INCO ERA-WIDE INARMERA-ICT project and its Pilot 2 that assisted to this research; and the SCS Armenia Thematic Grant 13-1B340, 2013-2015, which provided partial support to this research.

REFERENCES

[1] W. Blaschke, Kreis und Kugel, Berlin, 1956.
[2] H. Aslanyan and J. Rolim, Interference minimization in wireless networks, 3rd International Workshop on Wireless Network Algorithm and Theory. WiNA2010: 444-449 Hong Kong, China 2010.
[3] Franz Aurenhammer, Voronoi Diagrams – A Survey of a Fundamental Geometric Data Structure, ACM Computing Surveys, 23(3):345–405, 1991.
[4] L.Aslanyan, The isoperimetry problem and other related problems for the discrete spaces. Problemy Kibernetiki, Moscow, 36:85-127, 1979, (in Russian).
[5] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, “Integrated coverage and connectivity configuration in wireless sensor networks”, in Proceedings of the 1st international conference on Embedded networked sensor systems, ser. SenSys’03. New York, USA: ACM, 2003, pp.28–39.
[6] B. Chen, K. Jamieson, H. Balakrishnan, and R. Morris, “Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks,” Wireless Networks, vol. 8, pp. 481–494, 2002, 10.1023/A: 1016542229220.
[7] C.-F. Huang, Y.-C. Tseng, and H.-L. Wu, “Distributed protocols for ensuring both coverage and connectivity of a wireless sensor network,” ACM Trans. Sen. Netw., vol. 3, March 2007.
[8] Z. Zhou, S. Das, and H. Gupta, “Connected K-coverage problem in sensor networks,” in Proceedings of the 13th International Conference on Computer Communications and Networks, ser. ICCCN ’04, Chicago, IL, USA, Oct. 11–13, 2004, pp. 373–378.
[9] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” Ad Hoc Networks, vol. 7, no. 3, pp. 537 – 568, 2009.