Wireless Sensor Deployment for 3D Coverage with Constraints

Tycho Andersen and Srikanta Tirthapura
Dept. of Electrical and Computer Engineering, Iowa State University.
tanderse@iastate.edu, snt@iastate.edu

***Abstract*—We consider the problem of deploying wireless sensors in a three dimensional space to achieve a desired degree of coverage, while minimizing the number of sensors placed. Typical sensor deployment scenarios impose constraints on possible locations of the sensors, and on the desired coverage, but currently there is no unified way to handle these constraints in optimizing the number of sensors placed. We present a novel approach called *discretization* which allows us to cast the sensor deployment problem as a discrete optimization problem, and hence apply well-understood and flexible discrete optimization techniques for sensor deployment. Our results show that this approach yields solutions that nearly minimize the number of sensors used, while providing a high degree of coverage. Further, unlike typical approaches to sensor deployment, where 3D coverage is significantly more complex than 2D coverage, discretization is equally easy to apply for 2D as well as 3D coverage.**

I. Introduction

Many sensor networking applications require the placement of sensors such that a large fraction, and sometimes, all, of a "target region" is monitored by the sensors that are placed. Each sensor is able to monitor physical phenomena in a certain region around its location. This naturally leads to an optimization problem where the sensor locations should be chosen carefully so that the smallest number of sensors are used to monitor the region.

We faced a sensor deployment problem in the context of building a "smart" emergency evacuation system. The overall goal of our project is to build a network of wireless sensors which can provide informed guidance for evacuation in case of an emergency. The setup is as follows. Sensors are deployed throughout a building. Each sensor monitors the region around itself and detects regions of high temperature in its vicinity. This information is processed in the network, and is used to activate appropriate directions on "exit" signs throughout the building. As a result, evacuation instructions displayed on the exit signs will depend on the current conditions in the building, and this is potentially more useful than a traditional exit sign, which shows the same information regardless of current conditions. An important component of our project was to determine the locations where sensors would be deployed, to achieve the desired level of monitoring. While the sensor deployment problem has been widely studied in the literature (for example, [1], [2]), many requirements in our problem rendered the prior solutions unusable.

Firstly, the region to be covered by the sensors was three dimensional, such as the rooms and hallways of a building. Much of the research on sensor deployment, including [3]–[5] has focused on two dimensional coverage, and this is not applicable to our case. In our situation, there were restrictions on where sensors could be placed, such as: sensors could only be placed on the walls and ceilings of a corridor, they could not be placed on the ground, and could not be suspended in mid air. Even on the walls, there were some regions where sensors could not be placed, such as on doors or whiteboards. Previous work on sensor deployment for 3D coverage, such as [1], [2] are unable to handle such constraints – their approach returns sensor locations that could be anywhere in the target region. Further, in our setting, due to the presence of walls through which a sensor may or may not be able to sense, the region monitored by a sensor is usually not a sphere, but could be of a more complex shape. This motivated us to investigate more flexible techniques for sensor deployment, that could optimize the number of sensors placed in the presence of all the constraints described above. We first describe our problem more precisely before introducing our approach.

A. Problem

The sensor deployment problem can be framed as an optimization problem as follows. The number of sensors need to be minimized while certain constraints need to be met: (1)The target area is "sufficiently" covered by the sensors. (2)The sensors are placed in "valid" locations.

Our coverage criterion is the general k-coverage requirement [3], for some positive integer k. Suppose that R is a connected 3D target region that needs to be monitored. R could be of any shape, and is usually the union of cuboids in our setting. Suppose further that we are given a finite set of locations L where sensors can potentially be placed. Each potential sensor location ℓ can be considered as a set $R_\ell \subset R$ of all points in R that are monitored by placing a sensor at ℓ.

Definition 1: A location ℓ is said to *cover* a point $p \in R$ if $p \in R_\ell$. A set of locations S is said to k-cover a point $p \in R$ if there are at least k different locations in S that cover p. A set of locations S is said to k-cover a set of points $P \subseteq R$ if S k-covers each point in P.

The task is to find the set of locations $L' \subseteq L$ of the smallest cardinality such that L' k-covers R. In most cases, as we describe below, it is sufficient to have near-complete coverage, i.e. choose a set of locations L' such that L' k-covers all but a small fraction of R.

978-1-4244-6314-5/09/$26.00 ©2009 IEEE

B. Our Approach: Discretization

The above optimization problem has constraints that deal with complex geometric shapes, because the region covered by each sensor is a continuous region, and the union of many such regions is a complex geometric object. However, we do not know of appropriate tools to deal with such geometric objects. Our approach, called *discretization*, reduces the continuous optimization problem to a discrete optimization problem. This reduction is useful since we are now able to apply our wide selection of tools available for discrete optimization, which are vastly more flexible than the tools that we have for continuous optimization.

The basic idea is as follows. We first choose a finite set of points, say $G \subset R$ using a procedure called *Discretize*, to be described below. We then choose a set of locations $L' \subseteq L$ such that G is k-covered by L', and the size of L' is as small as possible. We then place sensors at the locations in L'. The latter problem is a generalization of the classic discrete set-cover problem. Though the set-cover problem is NP-hard, there exist good heuristics that yield near-optimal solutions, and we adapt these heuristics for our problem.

Benefits. The main advantage of discretization is that it allows us to use flexible discrete optimization techniques to solve the sensor deployment problem. More specifically, it has the following benefits: (1)The coverage regions R_ℓ need not always be spheres, as is generally assumed. They can be more complex shapes, for example, coverage could stop at a wall, or cross a wall, depending on the circumstance. The size and shape of R_ℓ could vary depending on the location ℓ. (2)The potential sensor locations (L) can be an input to the problem, and thus it is possible to easily handle restrictions on the sensor locations. (3)Three-dimensional coverage, which is considered a much harder problem than the two-dimensional version, can also be handled by this framework. (4)It is possible to compare the cost of the obtained solution with the optimal, since lower bound techniques are well developed for discrete optimization. We show that using the algorithms we studied, it was possible to get solutions where the number of sensors used was nearly the minimum possible.

Drawbacks. The main drawback of discretization is that it is not possible to guarantee k-coverage of the complete region R. In our technique, we place sensors so that all points within some subset $G \subset R$ are k-covered. Clearly, this does not guarantee that all of R is k-covered. Indeed, for any finite set of points $G \subset R$ it is possible to construct cases such that all points in G are k-covered, but there are points in R that are not. Thus, it is possible there are "coverage holes" due to the deployment recommended by the algorithm.

However, we found in our simulations that if the points G are chosen densely from R, k-coverage of G results in k-coverage of almost all of R. As a result, the total volume of the coverage holes is very small when compared with the volume of R. As we choose larger sets G, the coverage holes became smaller, and in our simulations the volume of the coverage holes was only 1-2 % of the total volume of the region. It is possible to make this number even smaller by increasing the size of G.

II. Related Work

We first begin with a review of literature on sensor deployment for 3D coverage. [4] distinguishes between two problems, the sensor coverage problem and the sensor deployment problem. Quoting from [4], "Given a sensor network, the coverage problem is to determine how well the sensing field [region] is monitored or tracked by sensors, while the deployment problem is to address how to place sensors into a sensing field [region] to meet certain coverage requirements." According to the above definition, ours is a deployment problem. The above work proposes a solution to the coverage problem, but does not address deployment.

[1] proposes a solution to the 3D deployment problem through packing the region by copies of a regular polyhedron with a small "volumetric coefficient". They consider various choices of polyhedrons, and conclude that packing by a truncated octahedron provides the maximum volumetric coefficient, and hence the minimum number of sensors. This approach cannot be easily adapted to meet our requirements, since it assumes that sensors can be placed anywhere in the target region, which is not true in our case. It also assumes that the coverage regions of different sensors are all spheres of the same volume. [6] argues that extending usual 2D coverage algorithms for 3D coverage may be difficult.

Another direction is to simultaneously ensure coverage as well as sensor connectivity [7], [8]. These works consider the 2D case, and show that if certain conditions hold between the sensing and the transmission radii, then coverage implies connectivity. Some other representative work on 2D coverage includes [9].

The goal of the work in [10] is to deploy sensors to guaranteeing "point coverage", which is the case when a only a finite number of points have to be monitored. The resulting optimization problem is solved using integer linear programming, using techniques similar to our methods. In contrast, we pursue a different goal. Our goal is to cover almost all of the continuous target region, and we use discretization followed by discrete optimization as a means to this goal. A similar approach is taken by [11].

III. The Deployment Algorithm

We first describe our procedure for discretization. The procedure $Discretize(R, d, \ell)$ returns a "representative" set of points within R that are further used by the set covering algorithms. The procedure takes two other parameters, d and ℓ, where d is a real number, and ℓ is a 3-tuple describing the coordinates of a single point within R. A 3-dimensional grid is constructed with a side length d along each dimension, starting with ℓ as one of the grid points. The procedure finally returns the set of all grid vertices that lie inside the region R. The overall algorithm for placement is given as Algorithm 1.

Algorithm 1: Sensor Deployment for k-coverage

1) Choose parameters d and ℓ. Parameter ℓ should be the coordinates of some point inside R. Let $G \leftarrow Discretize(R, d, \ell)$.
2) Use an discrete set cover algorithm with multiplicity k (described below) to find the smallest possible subset of L, say L', such that L' k-covers G.
3) Return L' as the set of locations to deploy the sensors.

A. Discrete Set Cover with Multiplicity k

After discretization, the inputs to the remaining problem are G, the result of the discretization, L, the potential sensor location, and the parameter k. The goal is to find the smallest sized subset of L that k-covers G. We call this problem "set cover with multiplicity k". For the case of $k = 1$, this reduces to the well known set cover problem. The set cover problem is known to be NP-complete, so it is unlikely that there are polynomial time algorithms that return an optimal set cover. Hence, the set cover with multiplicity k is also NP-hard, and it is unlikely there are polynomial time algorithms for this either. However, there are heuristics known for set cover, which perform well in practice, and we adapted these algorithms for our problem. We first discuss a strategy for finding a lower bound for a solution, using linear programming, and then briefly describe two algorithms we used.

1) Lower Bound: The set cover with multiplicity k can be written as an integer linear program. Given a set G to be covered, and L, the set of potential sensor locations, the integer program is as follows. There is a variable x_s corresponding to each potential sensor location $x \in L$. For point $g \in G$, let $S(g)$ denote the set of all sensors in L that cover g.

Integer Program ILP.

$$\begin{aligned} \text{minimize} \quad & \sum_{\ell \in L} x_\ell \\ \text{subject to} \quad & \\ \text{1. For each } g \in G \quad & \sum_{\ell \in S(g)} x_\ell \geq k \\ \text{2. For each } \ell \in L \quad & x_\ell \in \{0, 1\} \end{aligned}$$

Constraint 1 ensures that each point in G is covered by at least k selected locations. Constraint 2 is a set of constraints ensuring integrality of the x_ℓ variables. Note that this formulation does not help in solving our optimization problem, since integer programming is NP-hard in general. However, it leads to a way of finding a lower bound for the optimal solution through the LP-relaxation technique. We relax the integrality constraints in the integer program by allowing the variable x to take values $0 \leq x_\ell \leq 1$.

Since the feasible region of the LP-relaxation is a superset of the feasible region of ILP, the optimal solution to the LP-relaxation is smaller than or equal to the optimal solution to ILP. Thus the optimal solution to the LP serves as a lower bound to the optimal solution to the integer program, and helps us assess the quality of the solutions returned by the heuristics. We solved the linear programs with the help of the *CPLEX* software [12].

2) Algorithms: We evaluated three algorithms for this set cover problem with multiplicity k: (1) a simple greedy algorithm, (2) an algorithm based on Linear Programming and (3) a variation of the ITEG algorithm proposed in [13].

The first solution that we investigated is an adaption of the standard Randomized Greedy (RG) algorithm [14]. While the greedy algorithm was originally designed for the set cover problem, we modified it to account for k-coverage.

The most effective solution was an algorithm based on the ITerated Enhanced Greedy (ITEG) algorithm, proposed in [13]. ITEG is a heuristic solution to the set cover problem which showed excellent practical performance. We extended ITEG to handle set cover with multiplicity k, in a similar manner to the extension of randomized greedy above.

IV. Results

Volume Measurement. One challenge here was to measure the volume of the region that was k-covered. Since these regions are of a very irregular shape, it is not possible to compute the volumes analytically. Thus, we employed the "Monte-Carlo" method to estimate these volumes to within a high degree of accuracy. Let R' denote the set of all points $x \in R$ such that x was not k-covered by our chosen set. For region $\mathcal{T}$, let $v(\mathcal{T})$ denote the volume of $\mathcal{T}$. The method is as follows. Select n random points within R (with replacement), and let f' be the fraction of these points that were within R'. Then, $f' \cdot v(R)$ is an unbiased estimator of $v(R')$. By making n large enough, the estimator can be made as close to $v(R')$ as desired, with very high probability. For a more formal treatment of this technique, we refer the reader to [15].

Evaluation Methodology. We evaluated the algorithms through simulations, and considered the following metrics:

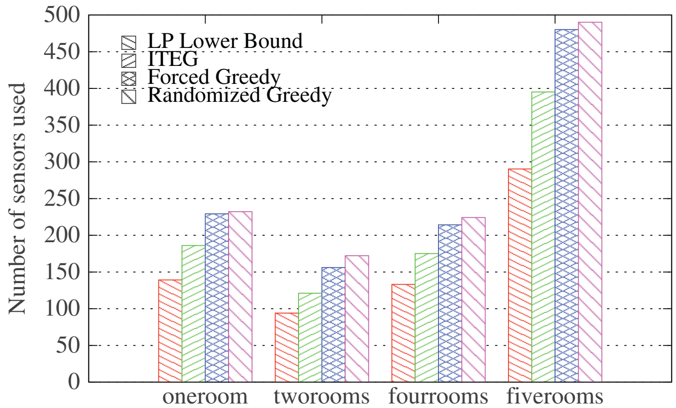
1) Cost: The number of sensors.
2) Coverage: The fraction of the input region that was k-covered by the solution that was returned.

We first evaluated how different algorithms performed on the input data sets. Then, we performed an in depth analysis of the "tworooms" input case to explore the influence of the two parameters, the coverage multiplicity k, and the grid spacing d. Specifically, we investigated the following questions.

1) How does the coverage vary with d?
2) How does the cost vary with d?
3) How does the cost vary with k?

We began by exploring how different algorithms performed on the input data sets, and then explored one input case in detail to determine the effects of the different parameters in our discretization. First, we present a comparison of the results of different algorithms on the input cases. For all the inputs, we allowed sensors to be placed anywhere in the region (not just on the walls), and we used a grid side length $d = 0.2$ for discretization. The results are shown in Figure 2 Four numbers are shown for each input: "LP lower bound" is a lower bound on the optimal number of sensors needed to cover the continuous region, obtained through the linear program.

Figure 2: All algorithms, compared with the lower bound.



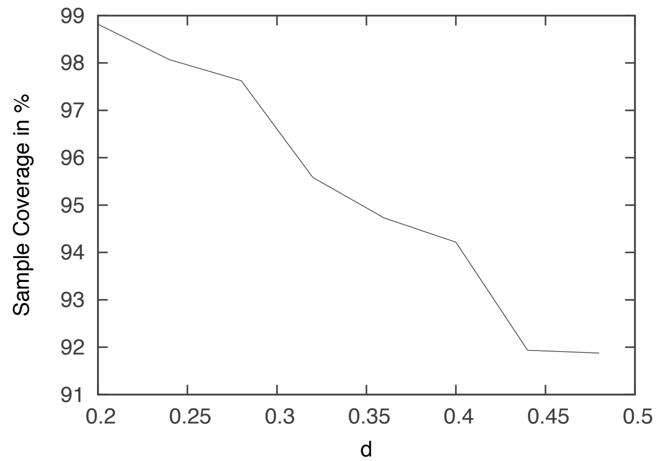
Impact of Grid Spacing d **on Coverage.** For the first question, Figure 3 shows the percentage of region covered, as a function of the grid spacing d. We experimented with d in the range $[0.2, 0.5]$, in steps of 0.01, and used sensors with a radius 1. For each value of d, we discretized the region with many different points for the origin ℓ in $Discretize(R, d, \ell)$, and the result shown is the average taken over all these different discretizations. The percentage of the region covered was computed using the Monte-Carlo method, as described in the beginning of this section. We used the ITEG algorithm with 10 iterations in all cases, since ITEG outperformed all other algorithms that we considered.

Figure 3 shows that even with $d = 0.5$, which is half the sensing radius, the coverage is 91 percent. When the grid spacing increased to a value close to the sensing radius, the coverage quickly deteriorated. For example, when the grid spacing was $d = 0.75$, which is 75% of the sensing radius, the coverage was only about 79%. From the above results, we can recommend that *the grid spacing* d *should be much smaller than the sensing radius, to get good coverage*.

V. Conclusion

We presented discretization, a flexible way to optimize sensor deployment in the presence of constraints such as restrictions on sensor locations, and non-uniform sensing regions. None of the previous approaches to sensor deployment could handle these constraints. Our flexibility comes at the cost of coverage holes. Through simulation, we found that the size of these coverage holes were as small as 1 % of the total volume in the cases we considered, and this number can be made even smaller with greater computational expense. For applications that work well with near-complete coverage, discretization is a compelling approach.

Figure 3: Coverage vs. Grid Spacing d



Acknowledgment: We thank Puviyarasan Pandian for his help in the initial phase of this work, and in exploring related work.

References

[1] S. M. N. Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional networks," in *Proc. Annual ACM International Conference on Mobile Computing and Networking (MOBICOM)*, 2006, pp. 346–357.

[2] N. Ahmed, S. S. Kanhere, and S. Jha, "Probabilistic coverage in wireless sensor networks," in *LCN '05: Proceedings of the The IEEE Conference on Local Computer Networks*, 2005, pp. 672–681.

[3] C.-F. Huang and Y.-C. Tseng, "The coverage problem in a wireless sensor network," in *WSNA '03: Proc. ACM International Conference on Wireless Sensor Networks and Applications*, 2003, pp. 115–121.

[4] C.-F. Huang, Y.-C. Tseng, and L.-C. Lo, "The coverage problem in three-dimensional wireless sensor networks," in *IEEE GlobeCom '04: Global Telecommunications Conference*, 2004, pp. 3182–3186.

[5] M. K. Watfa and S. Commuri, "A coverage algorithm in 3d wireless sensor networks," in *1st International Symposium on Wireless Pervasive Computing*, 2006, pp. 10–16.

[6] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme, "Sensor network configuration and the curse of dimensionality," in *The Third IEEE Workshop on Embedded Networked Sensors*, 2006.

[7] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated coverage and connectivity configuration in wireless sensor networks," in *SenSys '03: Proceedings of the 1st international conference on Embedded networked sensor systems*, 2003, pp. 28–39.

[8] H. Zhang and J. C. Hou, "Maintaining sensing coverage and connectivity in large sensor networks," *International Journal of Wireless Ad Hoc and Sensor Networks*, vol. 1, pp. 89–124, 2005.

[9] X.-Y. Li, P.-J. Wan, and O. Frieder, "Coverage in wireless ad hoc sensor networks," *IEEE Transactions on Computers*, vol. 52, no. 6, pp. 753–763, 2003.

[10] J. Wang and N. Zhong, "Efficient point coverage in wireless sensor networks," *Journal of Combinatorial Optimization*, vol. 11, no. 3, pp. 291–304, 2006.

[11] H. Q. Krishnendu Chakrabarty, S. Sitharama Iyengar and E. Cho, "Grid coverage for surveillance and target location in distributed sensor networks," *IEEE Transactions on Computers*, vol. 51, no. 12, pp. 1448–1453, 2002.

[12] "Ilog cplex," http://www.ilog.com/products/cplex/.

[13] E. Marchiori and A. Steenbeek, "An iterated heuristic algorithm for the set covering problem," 1998.

[14] U. Feige, "A threshold of ln *n* for approximating set cover," *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[15] M. Mitzenmacher and E. Upfal, *Probability and Computing*. Cambridge University Press, 2005.