Underwater Sensor Barriers with Auction Algorithms

Stanley Barr
Computer Science Department
University of Massachusetts Lowell
Lowell, MA 01854
Email: sbarr@cs.uml.edu

Benyuan Liu
Computer Science Department
University of Massachusetts Lowell
Lowell, MA 01854
Email: bliu@cs.uml.edu

Jie Wang
Computer Science Department
University of Massachusetts Lowell
Lowell, MA 01854
Email: wang@cs.uml.edu

Abstract

With current technologies submarines can thwart active or passive sonar detection. A viable alternative to detect submissible vessels is to use magnetic or acoustic sensors in close proximity to possible underwater pathways of them. This approach may require deploying large-scale underwater sensor networks to form barriers.

We show new results for construction barriers in 3D sensor networks. First, we prove that barriers are unlikely to exist in a large 3D fixed emplacement sensor field where sensor locations follow a Poisson point process. We then derive the notion of 3D *stealth distance* to measure how far a submarine can travel in a sensor network without detection. Finally, we describe energy conserving approaches for constructing a 3D barrier using mobile nodes to detect intruders. We focus on developing an energy efficient matching of mobile sensors that move to cover gridpoints using auction algorithms. We compare our results of the auction approach to an optimal approach using simulations and show that the auction algorithm produces similar results to the optimal approach at a reduced computational expense. This provides a fruitful new approach to constructing barriers in 3D sensor networks.

I. Introduction

International submarine sales on today's markets are not covered by any nonproliferation treaty, making it possible for nation states, organizations, or even individuals with sufficient resources to purchase submarines, some of which are capable of launching cruise missiles to deliver conventional, nuclear, chemical, or biological payloads [1]. Drug traffickers have also used submarines to smuggle illegal drugs to the US. The Washington Post reported that a total of thirteen submersible vessels of this kind were captured in 2007 [2], while many more were believed to have evaded US coastal patrols.

To make things worse, recent technology advances have also made it possible for submarines to evade standard sonar detection [3]. In particular, submarine hulls can be fitted with rubber anti-SONAR protection tiles to thwart active SONAR detection. These rubber tiles also dampen intra-submarine noise to thwart passive acoustic detection at a distance.

Thus, finding alternatives to detect submarines becomes important and timely. Using magnetic or acoustic enabled sensors in close proximity is a viable new approach. Large-scale deployments of underwater wireless sensor networks are expected to become a reality in the near future. Large-scale sensor deployments may occur by dispersal from an aircraft or artillery ordinance, as shown in Fig. 1. Sensors with different buoyancy may submerge at different depth. A sensor node may be equipped with a small ballast tank to control its weight and obtain different buoyancy. Such deployment strategies can deploy many sensors over a vast space quickly, but limit the control of sensor placement.

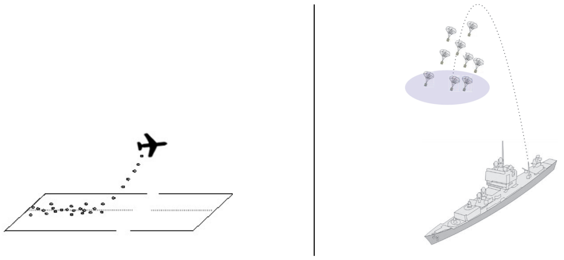

Fig. 1. Two sensor dispersal schemes

We want to form a barrier in a sensor network to detect moving objects. In 2-dimensional (2D) terrestrial strip sensor networks, a barrier is a chain of sensors from one end of the strip to the other end with overlapping sensing zones of adjacent sensors. No moving objects, regardless which paths they choose to pass through, can cross the chain undetected. In 3-dimensional (3D) underwater sensor networks, however, forming a barrier is much more subtle. For example, even if an unbroken chain of overlapping sensor zones between two opposing sides exists in a cuboid, intruders may still be able to pass over or under such a chain without being detected. Thus, an overlapping sensor chain no longer constitutes a barrier in a 3D space. Instead, a barrier in a 3D space should be a set of sensors with overlapping sensing zones of adjacent sensors that covers an entire (curly) surface that cuts across the space.

Early research on barrier sensor networks has been focused on 2-dimensional strips [4]–[11]. Underwater environments offer many challenges [12]. While there are some results for deployment, configuration, communications, and coverage in 3D environments there is still much work to be done [13]–[18]. Although not in the barrier context, it has been suggested that magnetometer enabled motes could detect moving submarines at distances of several hundred meters [19].

In this paper, we present the first set of results for constructing a barrier to detect intruding submarines in a 3D sensor network where sensor nodes are distributed uniformly at random following the Poisson Point Process. We show that a barrier in such a 3D sensor network for a finite density of sensors is unlikely to exist. This suggests the necessity to deploy sensors with at least limited mobility. How to minimize the energy consumed by the movement of underwater sensors is an important issue, as solar panels may not be applicable for submerged sensors.

We start by developing the notion of 3D *stealth distance*. This is the straight-line distance an intruder can move within a sensor field without being detected. We develop an energy-efficient approach to using mobile sensors to construct a barrier in a 3D space. First, we map sensors to grid positions using a classic centralized approach. Then, we turn to using an approximate solution based on auction theory to reduce the computation cost. Auctions provide a computationally feasible alternative to the optimal solution. We compare the performance characteristics of the auction based approach to the optimal approach. Our results show that the auction algorithm provides a computationally inexpensive and close approximation to the optimal solution.

This paper is organized as follows. In Section II, we develop our network model. In Section III, we show that the probability that a barrier exists in a randomly deployed 3D sensor networks is small and introduce the notion of stealth distance for 3D sensor networks. In Section IV, we describe our approach to mitigating issues raised in the previous section. In Section V, we describe our simulations and their results. We conclude in Section VI.

II. Network Model

We consider a sensor network consisting of sensors deployed in a large-scale 3D cuboid. For the initial configuration, we assume that the locations of these sensors are uniformly and independently distributed in the cuboid. Such a random initial deployment is desirable in scenarios where prior knowledge of the region of interest is not available. It may be the result of certain deployment strategies. Under this assumption, the sensor locations can be modeled by a stationary 3D Poisson point process, for some fixed sensor density λ.

We assume that each sensor can sense the environment and detect intruders in the 3D sphere of radius r. A point is said to be *covered* by a sensor if it is located in the sensing sphere of the sensor. The sensor network is thus partitioned into two parts: the covered space, which is the space covered by at least one sensor; and the uncovered space, which is the complement of the covered space. An intruder is said to be *detected* if its path lies within the covered space.

In Figure 2, each circle represents covered space. No matter wthat path an intruder takes moving from top to bottom he will be detected. Such a barrier is said to provide *strong barrier coverage*.

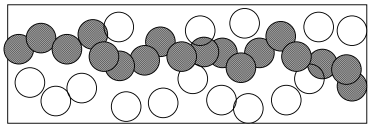

Fig. 2. Strong barrier coverage.

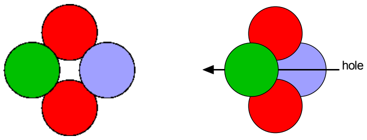

Fig. 3. Two different projections of the same sensors deployed in a 3D environment

III. Barrier Coverage in 3D Sensor Networks

In a 2D sensor network on a strip area, the barrier is directly related to the percolation of the network model, i.e., the existence and the number of disjoint connected sensor clusters that cross the strip. These connected sensor clusters act as “trip wires” that can detect any crossing intruders.

In a 3D sensor network, however, a sensor cluster connecting the opposite surfaces of the cuboid no longer constitutes a barrier that can detect crossing intruders. There could be “holes” in which an intruder can cross the cuboid undetected. Percolation of sensors no longer provides barrier coverage and intruders can evade via the uncovered space.

We consider a 3D cuboid of size $l \times w \times d$, where l, w, and d represents the length, width, and depth of the cuboid, respectively. A *crossing path* is a path that connects one surface of the cuboid to the opposite surface, where the ingress point and the egress point reside on two opposite surfaces of the cuboid. A crossing path is said to be covered if it intercepts at least one sensor. Intruders moving along covered crossing paths will be detected. A network provides a *barrier*, if any crossing path attempted by an intruder results in his being detected. A 3D sensor network is said to have a barrier in a specific direction if any crossing path intersecting the two surfaces perpendicular to the direction goes through the sensing zone of at least one sensor.

Without loss of generality, we assume that intruders attempt to penetrate the cuboid in the direction of depth. Clearly, strong barrier coverage requires a continuous surface that is fully covered by sensors. The following theorem characterizes the strong barrier coverage in 3D space.

Theorem 1: In a 3D cuboid of size $l \times w \times d$ where sensors are distributed according to a Poisson point process of density λ, there is a low probability of constructing a strong barrier coverage in the direction of depth for finite sensor density λ and depth d as $l, w \rightarrow \infty$.

Due to space constraints the proof has been omitted, but for complete details see [20].

In a wireless sensor network, as an intruder moves through the network, it will be detected when intersecting with the sensing sphere of a sensor. It is interesting to investigate the distance an intruder travels before first being detected by sensors. This distance, called the stealth distance and defined below, measures the intrusion detection performance of the sensor network.

Definition 1: Stealth Distance. Consider a 3D sensor network where sensors of sensing range r are distributed according to a Poisson point process of density λ. Assuming that an intruder is initially undetected and moves in a random direction along a straight line. The stealth distance of the intruder, X, is defined as the distance it travels before first detected by any sensor.

The notion of stealth distance was introduced in [5], and the authors derived an approximation of the expected stealth distance for 2D sensor networks. In the following theorem, we characterize the distribution of the stealth distance in a 3D sensor network. The distribution of the stealth distance in a 2D sensor network can be obtained similarly.

Theorem 2: In a 3D sensor network where sensors are distributed according to a Poisson point process of density λ and a node's sensing radius r the stealth distance of an intruder, X, follows an exponential distribution with parameter $\lambda\pi r^2$,

$$P(X < x) = 1 - e^{-\lambda\pi r^2 x}. \tag{1}$$

Once again, due to space constraints the proof has been omitted, but can be found in [20].

Based on Theorem 2, the expected stealth distance of a randomly located intruder is $E[X] = \frac{1}{\lambda\pi r^2}$, which is inversely proportional to the sensor density (λ) and the projected area of the sensing sphere (πr^2). Therefore, in order to shorten the stealth distance of an intruder, one can add more sensors, or use sensors with larger sensing range. To guarantee that the expected stealth distance of an intruder be smaller than a specific threshold l_0, we should have $\frac{1}{\lambda\pi r^2} < l_0$.

The above relationship between the stealth distance and the deployment parameters (sensor density and sensing range) provides important guidelines to the planning of sensor networks for intrusion detection.

IV. Moving Barriers

The results in the previous section show that barriers in a 3D network are unlikely to exist with uniformly deployed stationary emplacement sensors. To overcome this obstacle, we propose to use mobile sensors to form a barrier. To accomplish this task, we require that sensors be able to move in all three dimensions underwater, communicate with other sensors, localize their own positions, and detect intruders.

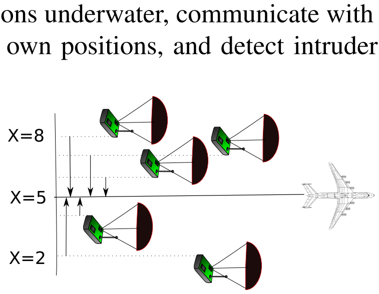

Fig. 4. Phase one: an air drop of sensors along path resulting in scattered placement along that path.

We note that sensors may be able to localize their own positions using a set of anchor nodes. Each of the anchor nodes is equipped with an underwater acoustic modem that submerges in the water, and a GPS that stays on the water surface. The GPS provides the location information of the node. An underwater sensor may communicate with a set of anchor nodes through their underwater acoustic modems and use the propagation delays of acoustic signals and the locations of the anchor nodes to calculate its location.

We propose a three phased approach for constructing a 3D barrier, which minimizes the max energy used by any sensor:

- Find an optimal vertical plane at which the sensors will form a grid-based barrier Figure 4.
- Compute each sensor's best grid assignment Figure 5.
- Move sensors from their initial dropped position directly to their final assigned grid positions in the optimal
- vertical plane.

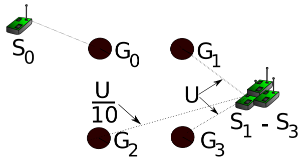

Fig. 5. Phase two example where sensors $S_1 - S_3$ derive the same utility U, from being assigned to G_1 or G_3. All receive a minute utility ($.1U$), from being assigned to G_2. S_0 does not have the energy to move to any grid point other than S_0. An auction occurs to assign $S_1 - S_3$ to $G_1 - G_3$.

A. Finding the optimal barrier location

Suppose as in Figure 4 an airplane drops a set of mobile sensors such that plane X is parallel to the coast line to be protected. The sensors must move to some plane X, such that X minimizes the energy expended by any one sensor. For this calculation, we are assuming that the sensors will only be moving in one direction to approach the X location.

Suppose X represents the optimal line for all sensors to approach. Let $d(x_i, x)$ be $|x - x_i|$ the distance traveled by any sensor s_i in the direction of X. We are attempting to minimize the distance traveled $min \sum_{i=1}^{n} d(x_i, x)$. To accomplish this let $x = (\max x_i - \min x_i)/2$, then the maximum distance any one sensor travels is minimized. In Figure 4, all the sensors will move to the line $X = 5$. It is straightforward to show that X must be somewhere inside the set x-coordinates of the sensors. As any two sensors move to meet one another, the total distance they travel is the distance between them. Thus, the sensors on either edge move the farthest, and half the distance between them minimizes the maximum distance any one sensor moves.

B. Optimal assignment of sensors to grid points

Assigning n sensors to n grid positions is related to the Assignment Problem, where we would seek a one-to-one matching between sensors and grid positions. The cost of each possible matching is the energy required to move the sensor to a grid point. A classic optimal solution is the Hungarian

Method and can be computed in $O(n^3)$, where n is the number of sensors [21]. It produces an one-to-one assignment which minimizes the total energy expended.

However, in our context, we are looking to minimize the maximal amount of energy drain any one sensor incurs moving to a grid position. This will maximize the network lifetime. Let k be the maximal amount of energy we allow any sensor to expend for moving to cover a grid position. We seek to find the smallest value of k such that each sensor expends at most k, yet all sensors can move to cover a grid position.

We can apply the Hungarian Method to compute k. Assume we have a function $HungarianK(S, G, k) \rightarrow Assignment, k'$. This function takes a list of sensor positions S, a list of grid positions G which are to be covered, a maximal value k, and computes a feasible assignment. If all the edges in the intermediate result do not exceed k then the assignment is returned otherwise the empty-set is returned. In addition k' represents the largest edge found in the assignment.

With this new function a centralized solution can be accomplished at a central node as follows. Sensors transmit positional information to a central node. Each sensor can be paired with every grid position resulting in a list of edge weights whose number is n^2. A binary search is conducted on the edge list and the midpoints of the edge weight list are fed as the parameter k to $HungarianK$. During the binary search, an admissible solution will result in $high$ being reduced otherwise low is increased. The running time for a solution will be $O(n^3 \log n)$.

C. Auction based assignment of sensors to grid points

The classic Hungarian method provides a centralized optimal solution to our assignment problem. To reduce computation cost, we present in this section a centralized approximate solution to the problem of assigning sensors to grid positions using *auction algorithms* [22]. Auction algorithms offer a reduced computational expense to the Hungarian method, while delivering close approximations to the optimal solution.

Auction algorithms arrive at an approximately optimal solution based on a set of agents bidding for the same resource, where each sensor labeled as i is bidding to be assigned to a particular grid position labeled as j. Each sensor associates a utility, which in our case is inversely proportional to a sensor's distance to a grid point. Let u_{ij} denote the utility of assigning sensor i to grid point j. Closer grid points have a higher utility to a sensor. Thus a sensor is willing to pay an increased price to compete for a grid point that maximizes its utility. The global known price for grid point j is denoted by p_j. The value $u_{ij} - p_j$ derives the benefit a sensor i achieves by being assigned to grid point j. Therefore, if the price at a particular grid point increases then a sensor will opt for a grid point with a lower utility and a higher benefit if it exists.

The algorithm starts by constructing an arbitrary assignment of sensors to grid positions. If each sensor derives the maximum benefit from its current assignment then the algorithm terminates. Otherwise, a sensor i seeks to be associated with j_i, which maximizes its benefit as shown in Eq 2.

Once the optimal candidate is identified, i raises p_{j_i} by the amount of $Increase$, which is the difference between the best benefit and the next best benefit. Now sensor i is associated with j_i and whoever j_i was associated with is associated with j. As j_i has the optimal benefit, $BestBen \geq NextBestBen$, therefore $Increase \geq 0$.

$$u_{ij_i} - p_{j_i} = \max_{j=1,\ldots,n} (u_{ij_i} - p_{j_i}) \tag{2}$$

$$p_{j_i} = p_{j_i} + Increase \tag{3}$$

$$Increase = BestBen - NextBestBen \tag{4}$$

$$BestBen = \max_{j} (u_{ij} - p_{j_i}) \tag{5}$$

$$NextBestBen = \max_{j \neq j_i} (u_{ij} - p_{j_i}) \tag{6}$$

It could turn out that $BestBen = NextBestBen$, i.e. a sensor derives the same benefit from being assigned to either of two separate grid positions. In that case, the bid increment would be zero. This could lead to repeated exchanges among a small group of sensors without increasing the price. To force convergence, we require that any bid must increase the price. To accomplish this in the algorithm, we replace Eq 4 with Eq 8 with a positive value ε. Further, we say that the system is in ε-equilibrium if each sensor is no more than ε off the maximum benefit possible. We accomplish this by substituting Eq 2 with Eq 7.

$$u_{ij_i} - p_{j_i} >= \max_{j=1,\ldots n} (u_{ij_i} - p_{j_i}) - \varepsilon \tag{7}$$

$$Increase = BestBen - NextBestBen + \varepsilon \tag{8}$$

We present an example to show how infinite loops might occur during the bidding process in phase two based on Eq 2 and Eq 4. Suppose that the sensors have limited movement so that S_0 can only bid for position G_0. Suppose that $S_1 - S_3$ are equidistant from G_1 and G_3. Then the sensors derive the same utility, denoted by U, from being assigned to G_1 or G_3. Suppose they can move all the way to G_2, but it would drain their energy reserves almost completely. Thus, G_2 offers a small non-zero utility to the sensors, which we denote by $.1 \times U$. Finally, suppose the initial mapping is $S_0 \rightarrow G_0, S_1 \rightarrow G_1, S_2 \rightarrow G_2, S_3 \rightarrow G_3$. To keep this example as simple as possible we remove S_0 and G_0 from our example. This configuration is depicted in Figure 5. The top two benefits are identical for all the sensors and the initial price for all grid points is zero. Therefore, sensors S_1, S_2, and S_3 will wind up endlessly swapping assignments.

If we substitute in Eq 7 and Eq 8 we see the prices associated with being assigned to a grid point increasing. During each round of bidding the price for either G_1 or G_3 is increased by at least ε. Let $m\varepsilon$ be the first price that exceeds the utility U. When such a price is reached for either grid point, the benefit $(U - m\varepsilon) < 0$. At this point, the small utility $(.1 \times U)$, offered by G_2 to any sensor and its zero price will cause a sensor to prefer to be assigned to G_2. Once a sensor chooses G_2, we will have a feasible assignment and the algorithm can terminate. As demonstrated, if a feasible assignment exists then this algorithm is guaranteed to terminate in a finite number of steps (see [22] for details). Once complete, sensors move from their deployment positions to their assigned grid positions.

V. Performance Evaluation

We use simulations to evaluate how the approximate solution would perform in a typical real world deployment and give some comparisons to the optimal solution. For evaluations regarding the optimal solution obtained from the Hungarian method, the reader is referred to our technical report [20].

The average ocean depth is 3,790m. Specialized submarines such as the Trieste can dive to 11,015m in the water [23]. Our simulations propose a sensor network that would be able to detect submarines up to 3,790m.

Recent papers have stated that commercial motes with magnetometers can detect submarines at distances of several hundred meters [19]. For our simulation, we chose a fixed detection range of 465m. A column of six sensors with the first being deployed at a depth of 320m allows coverage of 3840m. We allow for our sensors to sink from between 320m to 3520m. We assume that all sensors are deployed with identical energy reserves. The resulting cube's *depth* is 3200m. Modern submarines are capable of a submerged launch of anti-aircraft missiles [3]. This requires dispersal from a high flying plane to minimize the risk of being shot down during deployment. We allow sensors to drift away from the drop position with a radius up to 160m in any direction, so a cube's *width* is 320m. Finally, each sensor column is 660m apart. We assume that the aircraft uniformly drops sensors from the first column to the final column. The *length* of a cube ranges from 1920m to 55680m. Our simulations were run on an Intel Xeon ™ 3.20GHz CPU. The operating system was Red Hat Fedora Core 8. All software was written in Python.

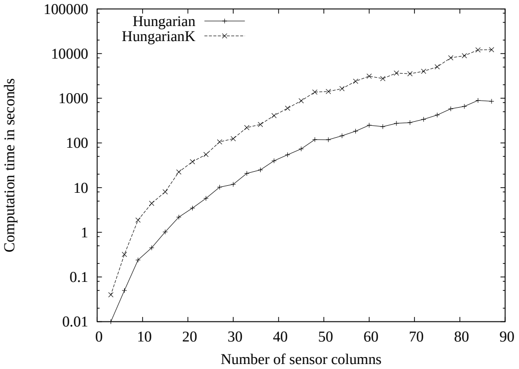

Fig. 6. Comparison of running time between the Hungarian and with our HungarianK variant.

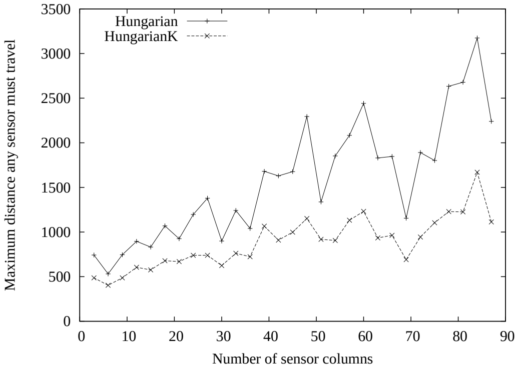

Fig. 7. Comparing the Hungarian Method with our HungarianK variant to determine maximum moving distance.

Figure 6 depicts our results. The x-axis represents the number of sensor columns being tested, which each sensor column being comprised of six sensors. The y-axis represents the total computation time required to achieve a result. This graphic shows that the computation time for computing HungarianK is within $O(n^4)$.

Figure 7 depicts the maximum movement any one sensor must travel to get to its optimal assignment. The x-axis is as before. The y-axis represents the maximum distance any one sensor must travel. As the network size increases the Hungarian Method requires move movement from sensors.

To seed the initial assignment during each run, we associated each sensor to an arbitrary grid position for all but the last experiment. This mapping was chosen to allow for an unbiased estimate of how quickly the system would converge. After some initial analysis, we chose five different values of ε to show how the system behaved. The values chosen were 0.5, 1.0, 2.0, 5.0, and 10.0. Each experiment computed a result using a single value of ε.

In the last experiment we started the search for a solution with $\varepsilon = 10.0$. Once a feasible solution was determined, the solution was used as an initial assignment to find a solution with $\varepsilon = 5.0$. This *annealing* process was continued until ε was reduced to a value of 0.5. Our simulations show how varying values of ε impact finding the optimal match between sensors and grid positions. We discuss different tradeoffs arising from varying the values of ε. The annealed auction is denoted in the figures by the lines $\varepsilon = Var$.

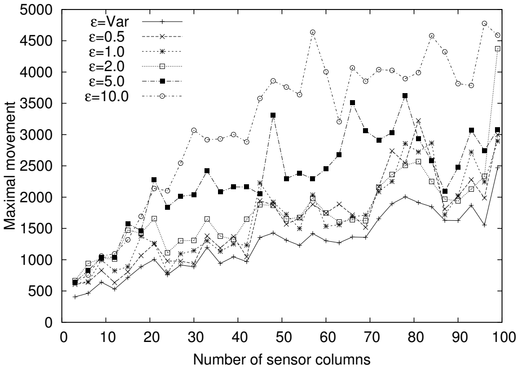

Fig. 8. Network size vs. Maximal movement required of individual sensor when varying ε .

In Figure 8, the x-axis is the same and the y-axis represents the maximum number of meters any sensor must travel. The smaller the value of ε, the shorter the travel distance for any sensor. In this case, the annealed process out performs all others. On average, the Hungarian method requires 1.34 times the distance of the annealed process and the modified Hungarian method with binary search, denoted by HungarianK, is .75 times better than the annealed process.

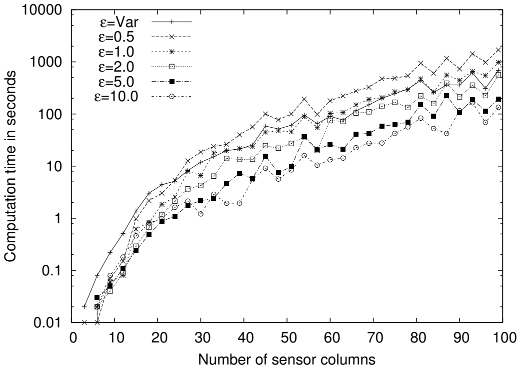

Fig. 9. Network size vs. computation time for the different values of ε

In Figure 9, the x-axis is the same and the y-axis represents the computation time of the approximate solution. For the annealed process, we see that the computation is more expensive when the number of sensor columns is smaller. As the number of sensor columns increase, the annealed process offers a noticable computational improvement over smaller values of ε. In general, the computation of the Hungarian method is comparable to when $\varepsilon = .5$. The HungarianK function grows $O(logn)$ faster making it difficult to simultaneously display all curves in a single graph.

VI. Conclusions and Future Work

In this work, we started by using an optimal centralized solution to construct a barrier. The solution resulted in the minimum energy being expended for construction. The solution provides a construction with minimum energy consumption. To improve scalability, we showed how an auction algorithm can produce results similar to the optimal while reducing the computation time required to produce such a result. Finally, the results show that using the annealing process further improves the convergence time and reduces the computational expense.

As discussed in the simulation section varying the value of ε trades computational complexity for looser approximations of the optimal value. However, this is partially dependent on the network size. It is still unclear how to choose good values of ε with respect to network size other than through simulation. This would would be very useful for large scale deployments.

The current work assumed homogeneous sensors as well as spherical detection ranges of those sensors. These assumptions are overly simplistic. In future work, we will be considering nodes with more realistic nodes. Nodes will have have irregular detection regions and a more probabilistic based sensing ability. We will consider nodes with different and possibility multiple sensing modalities as well as different processing capabilities. Constructing barriers from nodes like this should offer many challenging research opportunities.

Acknowledgements

We thank the anonymous reviewers who provided many useful comments. Benyuan Liu was supported in part by NSF under grant CNS-0721626. Jie Wang was supported in part by NSF under grants CNS-0709001 and CNF-0830314.

References

[1] (2008) Global submarine proliferation: Emerging trends and problems. [Online]. Available: http://www.nti.org/e_research/e3_74.html

[2] J. Forero, "Drug traffic beneath thewaves," in *Washington Post*, February 6, 2008, p. A04.

[3] J. Kong, J. Kong, J.-h. Cui, D. Wu, and M. Gerla, "Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications," in *Proc. IEEE Military Comm. Conf. MILCOM 2005*, J.-h. Cui, Ed., 2005, pp. 1535–1541 Vol. 3.

[4] D. Gage, "Command control for many-robot systems." in *Proc. of the Nineteenth Annual AUVS Tech. Sympos. (AUVS-92)*, 1992.

[5] C. Gui and P. Mohapatra, "Power conservation and quality of surveillance in target tracking sensor networks," 2004.

[6] S. Meguerdichian, F. Koushanfar, M. Potkonjak, and M. B. Srivastava, "Coverage problems in wireless ad-hoc sensor networks," in *Proc. IEEE Infocom*, 2001, pp. 1380–1387.

[7] S. Meguerdichian, F. Koushanfar, G. Qu, and M. Potkonjak, "Exposure in wireless ad-hoc sensor networks," in *ACM Mobile Computing and Networking*, 2001, pp. 139–150.

[8] P.-J. W. X.-Y. Li and O. Frieder, "Coverage in wireless ad-hoc sensor networks," *IEEE Transactions on Computers*, vol. 52(6), pp. 753–763, June 2003.

[9] G. Veltri, Q. Huang, G. Qu, and M. Potkonjak, "Minimal and maximal exposure path algorithms for wireless embedded sensor networks," in *Proc. of ACM Sensys*, 2003.

[10] T. H. L. S. Kumar and A. Arora, "Barrier coverage with wireless sensors," in *Proc. ACM Mobicom*, 2005.

[11] B. Liu, O. Dousse, J. Wang, and A. Saipulla, "Strong barrier coverage of wireless sensor networks," in *Proc. of Mobihoc '08*.

[12] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: research challenges," *Ad Hoc Networks (Elsevier*, vol. 3, pp. 257–279, 2005.

[13] ——, "Challenges for efficient communication in underwater acoustic sensor networks," *SIGBED Rev.*, vol. 1, no. 2, pp. 3–8, 2004.

[14] S. Poduri, S. Pattem, B. Krishnamachari, and G. S. Sukhatme, "Sensor network configuration and the curse of dimensionality," 2006.

[15] V. Ravelomanana, "Extremal properties of three-dimensional sensor networks with applications," 2004.

[16] H. M. Ammari and S. K. Das, "Joint k-coverage and hybrid forwarding in duty-cycled three-dimensional wireless sensor networks." pp. 170–178, 2008. [Online]. Available: http://dblp.uni-trier.de/db/conf/secon/secon2008.html#AmmariD08

[17] S. M. N. Alam and Z. J. Haas, "Coverage and connectivity in three-dimensional networks," pp. 346–357, 2006.

[18] D. Pompili, T. Melodia, and I. F. Akyildiz, "Deployment analysis in underwater acoustic wireless sensor networks," pp. 48–55, 2006.

[19] T. Nguyen, T. Nguyen, D. Nguyen, and D. Tran, "Stochastic binary sensor networks for noisy environments," in *Proc. First Internat. Conf. on Comm. and Electronics ICCE '06*, D. Nguyen, Ed., 2006, pp. 41–46.

[20] S. Barr, B. Liu, and J. Wang, "Barrier coverage for underwater sensor networks," in *Milcom 2008 Proceedings*.

[21] H. Kühn, "The Hungarian Method for the Assignment Problem," *Naval Research Logistics Quarterly*, vol. 2, pp. 83–97, 1955.

[22] D. Bertsekas, *The auction algorithm: A distributed relaxation method for the assignment problem.* Springer, 1988, vol. 14, no. 1.

[23] (2006) Seven miles down: The story of the bathyscaph trieste. [Online]. Available: http://bjsonline.com/watches/articles/0022_3.shtml