Optimizing Sensor Placement for Intruder Detection with Genetic Algorithms

Samuel R. Barrett

Abstract—**Sensor networks are effective tools for detecting intruders. However, the standard technique of placing sensors in a perimeter is not optimal. Using optimization techniques to determine sensor placement can improve the effectiveness of the sensor network. The optimization should take into account the environmental conditions and place sensors to take advantage of these conditions. Additionally, there are multiple objectives to consider in sensor placement, specifically the probability of detection and the time to detect. Genetic algorithms are capable of optimizing both objectives simultaneously, achieving the Pareto-optimal curve. This allows the designer of the network to specify a necessary value for one objective and get sensor placements that optimize the other objective. Compared to the standard perimeter configurations, the genetic algorithm networks perform significantly better with respect to both probability of detection and time to detect.**

Index Terms— **Genetic algorithms, Object detection, Sensor networks, Site security monitoring**

I. Introduction

SENSOR placement strongly affects the performance of a sensor network. Ideally, the placement of the sensors should optimize their effectiveness. However, when there are multiple objectives to consider, finding the optimal placement is not simple. The most common objectives are the probability of detection and the time to detect. Unfortunately, it is not possible to simultaneously maximize both objectives because higher values in one objective result in lower values in the other. However, it is possible to find the Pareto frontier, the curve of optimal trade-offs between the objectives. This allows the designer of the sensor network to specify a value that is necessary for one objective and from that get the optimal value for the other objective.

Environment plays a large role in the placement of sensors. When placing sensors in the Hudson River, current speeds play strongly affect where and at what speed divers will swim. Ignoring these environmental conditions is inefficient and can even make a sensor network ineffective because more sensors may be needed where currents allow divers to move faster than would be possible under their own power.

The standard sensor network configuration is that of a perimeter. Placing the sensors in a perimeter ensures that any object coming within a certain distance of the target must pass by at least one sensor. However, this either keeps the perimeter radius relatively small or requires a large number of sensors. While the perimeter is an intuitive configuration, is it superior to other configurations? Are there other configurations that would increase either the accuracy or the speed of detection?

One method for finding optimal configurations is genetic algorithms. Genetic algorithms explore a large solution space and are effective when considering multiple objectives simultaneously. Unlike many optimization techniques, genetic algorithms are capable of capturing the entire Pareto frontier in one run. Other techniques get a few solutions by linear combinations of the objectives followed and get the curve by interpolation. Genetic algorithms are capable of optimizing all the objectives at once and getting many solutions in one run.

Manuscript received January 2, 2007. This research was supported by the Office of Naval Research, grant N00014-05-1-00632.

S. R. Barrett is with Stevens Institute of Technology, Hoboken, NJ 07030 USA (email:sbarrett@stevens.edu)

II. Related Work

There has been relatively little research performed in the area of optimizing sensor placement. Sensor readings and the fusion of those readings into a fitness for the entire sensor network is complicated, so mathematically optimizing sensor placement is too difficult. Genetic algorithms are well suited for this optimization because they can search complicated functions and avoid getting stuck on local minima. Very little research has been done on the optimizing sensor placements in the area of intrusion detection.

Olague and Mohr [4] considered a similar problem when they attempted to optimize the placement of cameras to minimize the error in sensing position. They used genetic algorithms successfully for the optimization. Similarly, attempts to solve the sensor placement problem for diagnosing problems in plants have been made [7]. This research focused on optimizing the placement within a discrete number of possible positions rather than the continuous space. The work confirmed that genetic algorithms can be a effective way to search a large possibility space to optimize sensor placement.

Attempts have been made to optimize sensor placement in an environment that affects sensors' readings [2]. Dhillon and Chakrabarty introduced obstacles into the sensor field that affected the amount of information a sensor can receive and then attempted to optimize the placements of the sensors. Vickers, Stolkin, and Nickerson [8] also explored the effects of the environment on sensor placement and found that including environmental information allowed the sensors to be placed far more efficiently. They used the preference of divers to swim with the current rather than fighting against water currents and found significant improvement by including diver preferences about the environment in the optimization. The idea that sensor placements other than the perimeter configuration may be superior was researched by Olariu and Nickerson [5]. They found that axial networks of sensors are less obvious to intruders and that these networks are effective at detecting intruders.

Goldberg [3] explored using genetic algorithms for optimization and introduced the idea of using domination ranking as a method for moving towards the Pareto frontier. The non-dominated sorting genetic algorithm II (NSGA-II) extended this by using the distances between solutions to spread them out along the Pareto frontier [1]. Goldberg also describes many of the other useful techniques in genetic algorithms such as crossover and mutation techniques as well as fitness scaling.

The genetic algorithm implemented for this research was proposed by Deb, Pratap, Agarwal, and Meyarivan [1]. This genetic algorithm is an effective method for optimizing several objectives. Raisanen and Whitaker [6] successfully used NSGA-II to optimize a multiobjective problem and established that NSGA-II is a practical and efficient algorithm for multiobjective optimization.

III. Methods

The specific problem considered was attempting to optimize sensor placement to detect underwater intruders using acoustic sensors. The sensor readings were simulated using a probabilistic model where the probability of detection was directly proportional to $1/Range^2$. The environment used was simulated from the Hudson River near the New York City harbor. The water currents were interpolated from real sensor readings in the Hudson River. Currents vary during the day, especially in the Hudson River. To simplify the calculations this paper only considers the currents from one time of day. With more calculations, the networks could be evaluated for several environments.

The sensor networks were evaluated for probability of detection and time to detect. This evaluation consisted of sending a number of noise-creating diver objects towards a target. The divers attempted to head straight towards the target at each time step, but their trajectories were affected by the water currents, resulting in curved paths. Each sensor reported its probability of detection based on the shortest distance between the diver's trajectory and the sensor's position. Then, the sensors' probabilities of detection were combined to find the overall probability of detection. Finally, the probabilities were averaged, resulting in the average probability of detection for the sensor configuration. The time to detect was found by multiplying the probability that the intruder was not yet detected with both the probability that the intruder was just detected and the present time. Finally, the fitness of the network was taken to be the average probability of detection and the average time to detect.

IV. Results

The genetic algorithm resulted in a collection of sensor networks of varying effectiveness and using perimeter configurations with varying radii of perimeters created another collection of sensor networks. Each sensor network was evaluated using two objectives: probability of detection and time to detect. The figures show the comparisons of the genetic algorithm networks with the perimeter networks. For easier and more intuitive reading, the figures graph the probability of detection versus the inverse of the time to detect, or the time until target. The probability of detection is the average chance that a diver moving towards the target will be detected, and the time until target is the total time the intruder would take to reach the target minus the time when the intruder is detected on average. Both objectives are normalized so that the values vary between 0 and 1.

In the figures and calculations, the discovered networks shown are the non-dominated networks discovered by the genetic algorithm. They are the best of the networks the genetic algorithm produced and would likely survive and parent more networks in future generations. It is more effective to compare the networks with a single number representing the distance from (0,0) rather than the multiple

$(\text{Probability of Detection}) \propto 1/Range^2$

For $j = 1$ to (Number Of Divers)

  $(\text{Total Probability of Detecting})_j = 0$

  $(\text{Time to Detect})_j = 0$

  For $i = 1$ to (Number Of Sensors)

    $(\text{Time to Detect})_j = (1 - (\text{Time to Detect})_j) * (\text{Probability of Sensor i Detecting})_j * (\text{Time Passing near Sensor i})_j$

    $(\text{Total Probability of Detection})_j = ((\text{Total Probability of Detection})_j + (\text{Probability of Sensor i Detecting})_j) - ((\text{Total Probability of Detection})_j * (\text{Probability of Sensor i Detecting})_j)$

  end

  $(\text{Total Time to Detect}) = (\text{Total Time to Detect}) + (\text{Time to Detect})_j / (\text{Total Time})_j$

  $(\text{Total Probability of Detection}) \mathrel{+}= (\text{Total Probability of Detection})_j$

end

Figure 1. The calculations for determining the probability of detection and the time to detect for a sensor network.

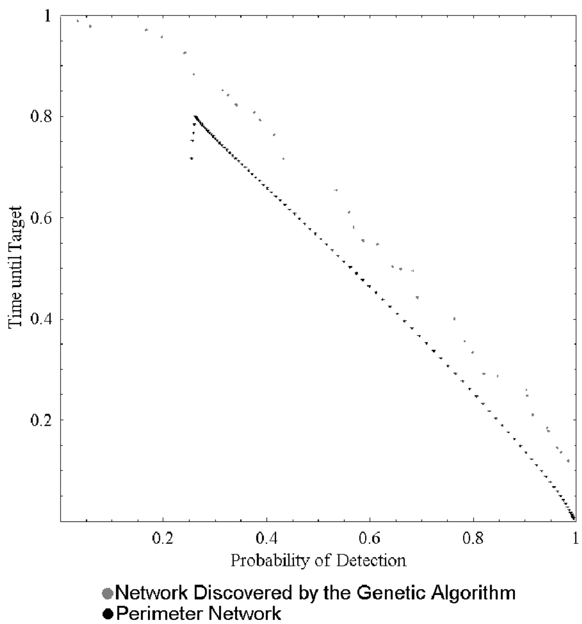

Figure 2. The comparison of the sensor networks reached by the genetic algorithms and the perimeter

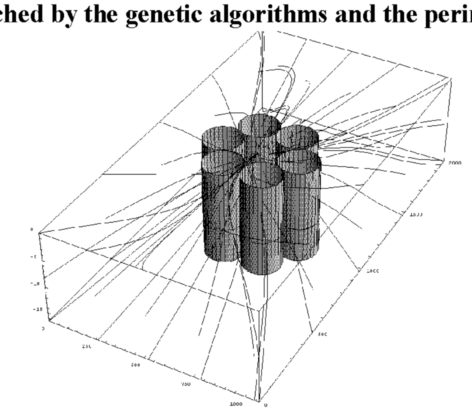

Figure 3. A network displaying the perimeter configuration. The cylinders represent the detection ranges of the sensors and the lines represent possible intruder

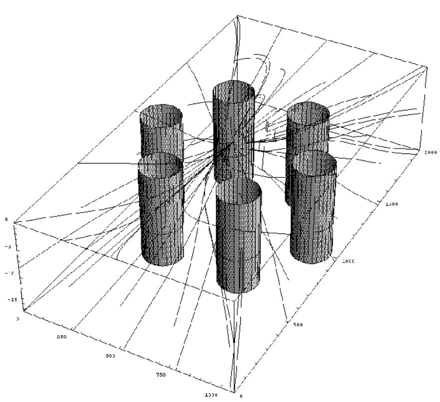

Figure 4. Another network displaying the perimeter configuration

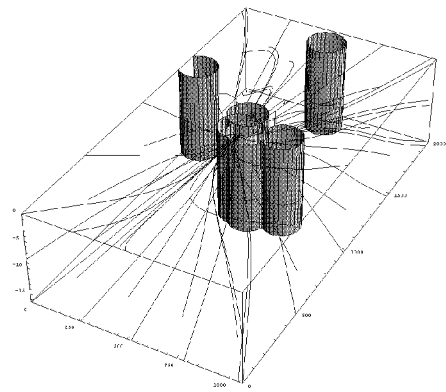

Figure 5. A sample network discovered by the genetic algorithm

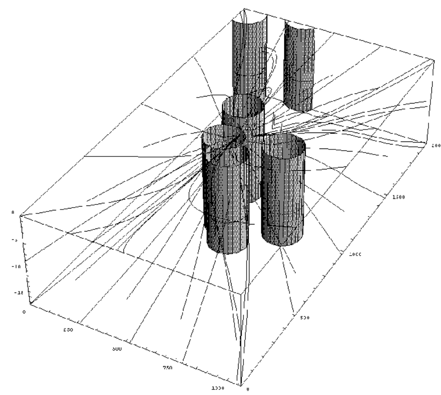

Figure 6. Another network discovered by the genetic algorithm.

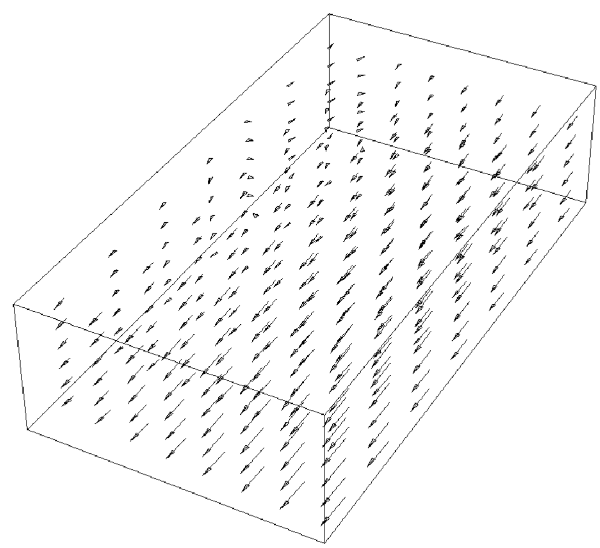

Figure 7. A sample of the current speeds and directions in the Hudson River.

objectives simultaneously. A larger distance from (0,0) implies the sensor network performs better because the point (0,0) represents a network incapable of detecting an intruder before it reaches the target. The mean distance from (0,0) for the networks discovered by the genetic algorithm is 0.904 while the mean distance from (0,0) for the perimeter networks is 0.837. There are a large number of discovered networks and perimeter networks, so the distance distribution can be modeled by the Gaussian distribution. The standard deviation of the discovered networks is 0.063 and the standard deviation of the perimeter networks is 0.080. Performing a z-test of the means results in a z-value of $5.103 > 3.090$. So the discovered networks' mean distance is larger than the perimeter networks' mean distance with significance level of 0.001 and a larger distance implies a better sensor network. A z-test of the medians results in a z-value of $6.275 > 3.090$, so the discovered networks' median distance is also larger than the perimeter networks' median distance with .0001 significance.

V. Discussion

The genetic algorithm reached solutions with higher times until target than any of the perimeter networks. The networks discovered by the genetic algorithm capture the entire continuum between high times until target and high probabilities of detection, while the perimeter configurations are more limited, only reaching a time until target of 0.800. This allows the designer of a sensor network to specify the value for any one objective they desire and then receive an optimal sensor network that will maximize the other objective.

Another advantage of the networks discovered by the genetic algorithm is the randomness of the sensor placements. In perimeter networks, it is relatively easy to discover where the sensors are placed. Sending several decoys through the network and seeing where they are discovered allows the prediction of the sensor placements. This allows future intruders to avoid the sensors or at least minimize their exposure to the sensors. Furthermore, if the intruders know where the sensors are placed, they may be able to disable the sensors. It is very difficult to accurately predict the positions of the sensors for the networks discovered by the genetic algorithm. Sending many decoys may result in the discovery of the sensor placements, but this many decoys would raise suspicion and limit the effectiveness of an intruder. Minimizing the knowledge given to the intruders maximizes the effectiveness of the sensor network.

It is very complicated to accurately measure randomness. However, the perimeter configurations are very regular, making them easy to predict. On the other hand, the discovered networks are irregular, making them hard to predict. By viewing the examples of the sensor placements, the reader can easily find the pattern in the perimeter networks: the sensors are evenly spaced a constant distance from the target. However, with the discovered networks, there is no clear pattern that would allow an intruder to predict the placement of the sensors.

In figures 3, 4, 5, and 6, the cylinders represent the maximum detection range of the sensors. The curved black lines represent the trajectories of several intruders attempting to move towards the target and being affected by the water currents. Figure 7 displays these water currents interpolated from readings in the Hudson River. Figures 3 and 4 are examples of the perimeter configurations with different distances from the target. Figures 5 and 6 depict sample networks reached by the genetic algorithm. The perimeter networks are less effective due to the need to either have sensors only in close proximity to the target or leave large gaps in the perimeter.

VI. Conclusions

The genetic algorithm was an effective optimization tool; the sensor networks it reached were superior in both objectives to those of the standard perimeter configuration. Furthermore, the genetic algorithm captured solutions further to the extremes of the objectives than the perimeter technique reached, giving a sensor network designer more options.

Many sensor placement algorithms ignore the environment, but this yields less effective networks. Considering the environmental conditions like water currents improved the effectiveness of the optimization because the environment can strongly influence the results of a sensor network.

Another advantage of the genetic algorithm's solutions is their randomness. A perimeter configuration gives away information about the location of the sensors which allows intruders to avoid or destroy sensors. The genetic algorithm's solutions, on the other hand, are random and therefore it is very hard for the intruder to predict the placement of the sensors. If the locations of the sensors are unknown, intruders are unable to avoid them.

References

[1] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivian, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation.* vol 6, pp. 182-197, April 2002.

[2] S. S. Dhillon and K. Chakrabarty. "Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks," *IEEE Wireless Communications and Networking Conference (WCNC '03).* vol. 3, pp. 1609-1614, March 2003.

[3] D. E. Goldberg, Genetic *Algorithms in Search, Optimization, and Machine Learning.* Reading, MA: Addison-Wesley, 1989.

[4] G. Olague and R. Mohr, "Optimal 3D Sensors Placement to Obtain Accurate 3D Points Positions"

[5] S. Olariu and J. V. Nickerson, "Protecting with Sensor Networks: Perimeters and Axes," *Proceedings of Military Communication Conference 2005.*

[6] L. Raisanen and R. M. Whitaker, "Comparison and Evaluation of Multiple Objective Algorithms for the Antenna Placement Problem," *Mobile Networks and Applications*. vol. 10, pp. 79-88, Feb. 2005.

[7] S. Spanache, T. Escobet, and L. Travé-Massuyès, "Sensor Placement Optimisation Using Genetic Algorithms," *Proceedings DX-2004*, 15th International Workshop on Principles of Diagnosis, Carcassonne, France June 2004.

[8] L. Vickers, R. Stolkin, and J. V. Nickerson, "Computational Environmental Models Aid Sensor Placement Optimization," Situation Management workshop at the Military Communications Conference, *IEEE MILCOM Proceedings*, October 2006.